WaveCluster with Differential Privacy

Ling Chen [#1], Ting Yu [#1,2], Rada Chirkova [#1]

[#1] *Department of Computer Science, North Carolina State University, Raleigh, USA*
[#2] *Qatar Computing Research Institute, Doha, Qatar*
[1] lchen10@ncsu.edu, [1,2] tyu@{ncsu.edu,qf.org.qa}, [1] rychirko@ncsu.edu

***Abstract*—WaveCluster is a important family of grid-based clustering algorithms that are capable of finding clusters of arbitrary shapes. In this paper, we investigate techniques to do WaveCluster while ensuring differential privacy. Instead of taking a case-by-case approach (as WaveCluster can be instantiated with any wavelet transform), our goal is to develop general techniques that can be applied to any instantiated WaveCluster algorithm. We show that straightforward techniques based on synthetic data generation and introduction of random noise when quantizing the data, though generally preserving the distribution of data, often introduce too much noise to preserve useful clusters. We then propose two optimized techniques, PrivTHR and PrivTHR$_{EM}$, which are independent of the specific Wavelet transform and can significantly reduce data distortion during two key steps of WaveCluster: the quantization step and the significant grid identification step. We conduct extensive experiments based on three large synthetic datasets, and show that PrivTHR and PrivTHR$_{EM}$ achieve high utility when privacy budgets are properly allocated.**

I. Introduction

Clustering is an important class of data analysis that has been extensively applied in a variety of fields, such as identifying different groups of customers in marketing and grouping homologous gene sequences in biology research [1]. Clustering results allow data analysts to gain valuable insights into data distribution when it is challenging to make hypotheses on raw data. Among various clustering techniques, a grid-based clustering algorithm called WaveCluster [2], [3] is famous for detecting clusters of arbitrary shapes. WaveCluster relies on wavelet transforms, a family of convolutions with appropriate kernel functions, to convert data into a transformed space, where the natural clusters in the data become more distinguishable. WaveCluster provides a framework that allows any kind of wavelet transform to be plugged in for data transformation, such as the Haar transform [4] and Biorthogonal transform [5].

In many data-analysis scenarios, when the data being analyzed contains personal information and the result of the analysis needs to be shared with the public or untrusted third parties, sensitive private information may be leaked, e.g., whether certain personal information is stored in a database or has contributed to the analysis. Consider the databases *A* and *B* in Figure 1. These two databases have two attributes, *Monthly Income* and *Monthly Living Expenses*, and the records differ only in one record, *u*. Without *u*'s participation in database *A*, WaveCluster identifies two separate clusters, marked by *blue* and *red*, respectively. With *u*'s participation, WaveCluster identifies only one cluster marked by color *blue* from database *B*. Therefore, merely from the number of clusters returned

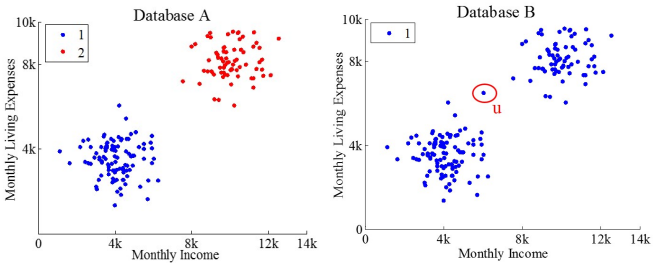


Fig. 1: Example of personal privacy breach in cluster analysis

(rather than which data points belong to which cluster), an adversary may infer a user's participation. Due to such potential leak of private information, data holders may be reluctant to share the original data or data-analysis results with each other or with the public.

In this paper, we develop techniques to perform WaveCluster clustering with differential privacy [6], [7]. Differential privacy provides a provable strong privacy guarantee that the output of a computation is insensitive to any particular individual. In other words, based on the output, an adversary has limited ability to make inference about whether an individual is present or absent in the dataset. Differential privacy is often achieved by the perturbation of randomized algorithms, and the privacy level is controlled by a parameter ϵ called "privacy budget". Intuitively, the privacy protection via differential privacy grows stronger as ϵ grows smaller.

A straightforward idea to enforce differential privacy in WaveCluster is to perturb the results obtained from the core wavelet transform. Such an approach requires careful sensitivity analysis for each single targeted wavelet transform, i.e., the maximum change caused to the wavelet transform output when any individual record in the input data is modified. In the big family of wavelet transforms, some transforms are rather complex, requiring case-by-case non-trivial efforts to derive their sensitivity, which is not an extensible solution. In this paper, instead, we aim to develop a general technique to achieving differential privacy on WaveCluster, which would be independent of wavelet transform.

We first consider a general technique, Baseline, that adapts existing differentially private data-publishing techniques to WaveCluster through synthetic data generation. Specifically, we could generate synthetic data based on any data model of the original data that is published through differential privacy, and then apply WaveCluster using any wavelet transform

over the synthetic data. Baseline seems particularly promising as many effective differentially private data-publishing techniques have been proposed in the literature, all of them striving to preserve important properties of the original data. Therefore, hopefully the "shape" of the original data is also preserved in the synthetic data, and consequently could be discovered by WaveCluster. In this way, we bypass the sensitivity analysis of wavelet transform. Unfortunately, as we will show later in the paper, this synthetic data-generation technique often cannot produce accurate results. Differentially private data-publishing techniques such as spatial decompositions [8], adaptive-grid [9], and Privelet [10], output noisy descriptions of the data distribution and often contain negative counts for sparse partitions due to random noise. These negative counts do not affect the accuracy of large range queries (which is often one of the main utility measures in private data publishing) since zero-mean noise distribution smoothes the effect of negative counts. However, negative counts cannot be smoothed away in the synthesized dataset, which are typically set to zero counts. As will be shown in Section VI, such synthetic data generation significantly distorts the data distribution and reduces the accuracy of the WaveCluster results.

Motivated by the above challenge, we propose three techniques that enforce differential privacy on the key steps of WaveCluster, rather than relying on synthetic data generation. WaveCluster accepts as input a set of data points in a multi-dimensional space, and consists of the following main steps. First, in the quantization step WaveCluster quantizes the multi-dimensional space by dividing the space into grids, and computes the count of the data points in each grid. These counts of grids form a count matrix M. Second, in the wavelet transform step WaveCluster applies wavelet transform on the count matrix M to obtain the approximation of the multi-dimensional space. Third, in the significant grid identification step WaveCluster identifies significant grids based on the pre-defined density threshold value. Fourth, in the cluster identification step WaveCluster outputs as clusters the connected components from these significant grids [11]. To enforce differential privacy on WaveCluster, we first propose a technique, PrivQT, that introduces Laplacian noise to the quantization step. However, such straightforward privacy enforcement cannot produce usable private WaveCluster results, since the noise introduced in this step significantly distorts the density threshold for identifying significant grids. To address this issue, we further propose two techniques, PrivTHR and PrivTHR$_{EM}$, which enforce differential privacy on both the quantization step and the significant grid identification step. These two techniques differ in how to determine the noisy density threshold. We show that by allocating appropriate budgets in these two steps, both techniques can achieve differential privacy with significantly improved utility.

Traditionally, the effectiveness of WaveCluster is evaluated through visual inspection by human experts (i.e., visually determining whether the discovered clusters match those reflected in the user's mind). Unfortunately, as visual inspection is not quantitative, this imposes challenges when assessing the utility of differentially private WaveCluster. Therefore, it is hard to systematically compare through visual inspection the impact of different techniques. Generally, researchers use quantitative measures to assess the utility of differentially private results, such as relative or absolute errors for range queries and prediction accuracy for classification. But there is no existing utility measures for density-based clustering algorithms with differential privacy.

Given true and differentially private WaveCluster results, these measures measure the dissimilarity of the significant grids and the clusters of significant grids, which are the outputs of two key steps in WaveCluster, significant grid identification and cluster identification.

To mitigate this problem, in this paper we propose two types of utility measures. The first is to measure the dissimilar significant grids and clusters differences between true and private WaveCluster results, which correspond to the outputs of the two key steps in WaveCluster, significant grid identification and cluster identification. To more intuitively understand the usefulness of discovered clusters, our second utility measure considers one concrete application of cluster analysis, i.e., to build a classifier based on mined clusters, and then use that classifier to predict future data. Therefore the prediction accuracy of the classifier from one aspect reflects the actual utility of private WaveCluster.

To evaluate the proposed techniques, our experiments use three synthetic datasets, which contain different data shapes that are especially interesting in the context of clustering. Our results confirm that PrivTHR and PrivTHR$_{EM}$ enforce differential privacy in both the quantization step and the significant grid identification step, achieve high utility, and are superior to Baseline and PrivQT.

II. Related Work

The focus of initial work on differential privacy [7], [12]–[15] concerned the theoretical proof of its feasibility on various data analysis tasks, e.g., histogram and logistic regression.

More recent work has focused on practical applications of differential privacy for privacy-preserving data publishing. An approach proposed by Barak et al. [16] encoded marginals with Fourier coefficients and then added noise to the released coefficients. Hay et al. [17] exploited consistency constraints to reduce noise for histogram counts. Xiao et al. [10] proposed *Privelet*, which uses wavelet transforms to reduce noise for histogram counts. Cormode et al. [8] indexed data by *kd*-trees and *quad*-trees, developing effective budget allocation strategies for building the noisy trees and obtaining noisy counts for the tree nodes. Qardaji et al. [9] proposed uniform-grid and adaptive-grid methods to derive appropriate partition granularity in differentially private synopsis publishing. Xu et al. [18] proposed the *NoiseFirst* and *StructureFirst* techniques for constructing optimal noisy histograms, using dynamic programming and Exponential mechanism. These data publishing techniques are specifically crafted for answering range queries. Unfortunately, synthesizing the dataset and applying WaveCluster on top of it render WaveCluster results useless,

since these differentially private data publishing techniques do not capture the essence of WaveCluster and introduce too much unnecessary noise for WaveCluster.

Another important line of prior work focuses on integrating differential privacy into practical data analysis tasks, such as regression analysis, model fitting, classification and etc. Chaudhuri et al. [19] proposed a differentially private regularized logistic regression algorithm that balances privacy with learnability. Zhang et al. [20] proposed a differentially private approach for logistic and linear regressions that involve perturbing the objective function of the regression model, rather than simply introducing noise into the results. Friedman et al. [21] incorporated differential privacy into several types of decision trees and subsequently demonstrated the tradeoff among privacy, accuracy and sample size. Using decision trees as an example application, Mohammed et al. [22] investigated a generalization-based algorithm for achieving differential privacy for classification problems.

Differentially private cluster analysis has also be studied in prior work. Zhang et al. [23] proposed differentially private model fitting based on genetic algorithms, with applications to *k*-means clustering. McSherry [24] introduced the PINQ framework, which has been applied to achieve differential privacy for *k*-means clustering using an iterative algorithm [25]. Nissim et al. [26] proposed the sample-aggregate framework that calibrates the noise magnitude according to the smooth sensitivity of a function. They showed that their framework can be applied to *k*-means clustering under the assumption that the dataset is well-separated. These research efforts primarily focus on centroid-based clustering, such as *k*-means, that is most suited for separating convex clusters and presents insufficient spatial information to detect clusters with complex shapes, e.g. concave shapes. In contrast to these research efforts, we propose techniques that enforce differential privacy on WaveCluster, which is not restricted to well-separated datasets, and can detect clusters with arbitrary shapes.

III. Preliminaries

In this section, we first present the background of differential privacy. Then we depict WaveCluster algorithm followed by our problem statement.

A. Differential Privacy

Differential privacy [6], [12] is a recent privacy definition, which guarantees that an adversary cannot infer an individual's presence in a dataset from the randomized output, despite having knowledge of all remaining individuals in the dataset.

Definition 1: **(ϵ-differential privacy)**: Given any pair of neighboring databases D and D' that differ only in one individual record, a randomized algorithm A is ϵ-differentially privacy iff for any $S \subseteq Range(A)$:

$$Pr[A(D) \in S] \leq Pr[A(D') \in S] * e^{\epsilon}$$

The parameter ϵ indicates the level of privacy. Smaller ϵ provides stronger privacy. When ϵ is very small, $e^{\epsilon} \approx 1+ \epsilon$. Since the value of ϵ directly affects the level of privacy, we refer to it as the *privacy budget*. Appropriate allocation of the privacy budget for a computational process is important for reaching a favorable trade-off between privacy and utility. The most common strategy to achieve ϵ-differential privacy is to add noise to the output of a function. The magnitude of introduced noise is calibrated by the privacy budget ϵ and the sensitivity of the query function. The sensitivity of a query function is defined as the maximum difference between the outputs of the query function on any pair of neighboring databases.

Definition 2: **(Sensitivity)**: The sensitivity S of a query function f is :

$$\Delta f = \max_{D,D'} \parallel f(D) - f(D') \parallel_1$$

There are two common approaches to achieve ϵ-differential privacy: Laplace mechanism [7] and Exponential mechanism [27].

Laplace Mechanism: The output of a query function f is perturbed by adding noise from a Laplace distribution with probability density function $f(x|b) = \frac{1}{2b}\exp(-\frac{|x|}{b})$, $b = \frac{\Delta f}{\epsilon}$. The following randomized mechanism A_l satisfies ϵ-differential privacy:

$$\mathcal{A}_l(D) = f(D) + Lap(\frac{\Delta f}{\epsilon})$$

Exponential Mechanism: This mechanism returns an output that is close to the optimum, with respect to a quality function. A quality function $q(D, r)$ assigns a score to all possible outputs $r \in R$, where R is the output range of f, and better outputs receive higher scores. A randomized mechanism A_e that outputs $r \in R$ with probability

$$Pr[\mathcal{A}_e(D) = r] \propto exp(\frac{\epsilon q(D, r)}{2S(q)})$$

satisfies ϵ-differential privacy, where $S(q)$ is the sensitivity of the quality function.

Differential privacy has two properties: sequential composition and parallel composition. Sequential composition is that given n independent randomized mechanisms $A_1, A_2, \ldots, A_n$ where A_i $(1 \leq i \leq n)$ satisfies ϵ_i-differential privacy, a sequence of A_i over the dataset D satisfies ϵ-differential privacy, where $\epsilon = \sum_1^n(\epsilon_i)$. Parallel composition is that given n independent randomized mechanisms $A_1, A_2, \ldots, A_n$ where A_i $(1 \leq i \leq n)$ satisfies ϵ-differential privacy, a sequence of A_i over a set of disjoint data sets D_i satisfies ϵ-differential privacy.

B. WaveCluster

WaveCluster is an algorithm developed by Sheikholeslami et al. [2], [3] for the purpose of clustering spatial data. It works by using a wavelet transform to detect the boundaries between clusters. A wavelet transform allows the algorithm to distinguish between areas of high contrast (high frequency components) and areas of low contrast (low frequency components). The motivation behind this distinction is that within a cluster there should be low contrast and between clusters there

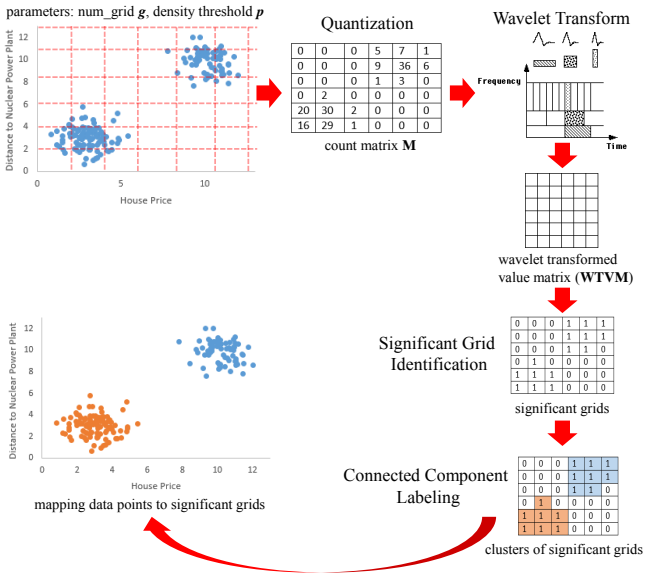


Fig. 2: Illustration of WaveCluster

should be an area of high contrast (the border). WaveCluster has the following steps, as shown in Figure 2:

Quantization: Quantize the feature space into grids of a specified size, creating a count matrix M.

Wavelet Transform: Apply the wavelet transform to the count matrix M, such as Haar transform [4] and Biorthogonal transform [5], and decompose M to the average subband that gives the approximation of the count matrix and the detail subband that has the information about the boundaries of clusters. We refer to the average subband as the wavelet-transformed-value matrix ($WTVM$).

Significant Grid Identification: Identify the significant grids from the average subband $WTVM$. WaveCluster constructs a sorted list of the positive wavelet transformed values obtained from $WTVM$, and computes d based on the p percentile density value in the sorted list, where p is a parameter given to WaveCluster. A grid whose wavelet transformed value is above d is considered as a significant grid. The data points in the non-significant grids are considered as noise.

Cluster Identification: Identify the connected components (two grids are connected if they are adjacent) as clusters from the significant grids using connected component labeling algorithm [11], map the clusters back to the original multi-dimensional space, and label the data points based on which cluster the data points reside.

In WaveCluster, users need to specify four parameters:

num_grid (g): the number of grids that the multi-dimensional space is partitioned into along one dimension. This parameter controls the scaling of quantization. Inappropriate scaling can cause problems of over-quantization and under-quantization, affecting the accuracy of clustering [3].

density threshold (p): a percentage value p ($0 \leq p \leq 100$) used to obtain the absolute density threshold value d. d is used to compare with the values in $WTVM$ for determining whether a grid is a significant grid. It is difficult to specify d directly since it is affected by the convolution of wavelet transform. Therefore, users normally specify a percentage value p and d is computed as the pth percentile of the positive values in $WTVM$.

level: a wavelet decomposition level, which indicates how many times the wavelet transform will be applied. The larger the level is, the more approximate the result will be. In our techniques, we set level to 1 since a smaller level value provides more accurate results [3].

wavelet: the wavelet transform to be applied. Haar transform [4] is one of the simplest wavelet transform and widely used, which is computed by iterating difference and averaging between odd and even samples of a signal (or a sequence of data points). Other commonly used wavelet transforms include Biorthogonal transform [5].

C. Motivating Scenario and Problem Statement

Consider a scenario with two participants: the data owner (e.g. hospitals) and the querier (e.g. data miner). The data owner holds raw data and has the legal obligation to protect individuals' privacy while the querier is eager to obtain cluster analysis results for further exploration. The goal of our work is to enable the data owner to release cluster analysis results using WaveCluster while not compromising the privacy of any individual who contributes to the raw data. The data owner has a good knowledge of the raw data and is able to provide appropriate parameters (e.g. num_grid and density threshold) for WaveCluster. We next give our problem statement.

Problem Statement. Given a raw data set D, appropriate WaveCluster parameters for D and a privacy budget ϵ, our goal is to investigate effective approach A such that A (1) satisfies ϵ-differential privacy, and (2) strikes an effective balance between the amount of introduced noise and the utility of the WaveCluster results.

IV. Approaches

In this section, we present four techniques for achieving differential privacy on WaveCluster, which are independent of wavelet transforms. We first describe the Baseline technique that achieves differential privacy through synthetic data generation. We then describe three techniques that enforces differential privacy on the key steps of WaveCluster.

A. Baseline Approach (Baseline)

A straightforward technique to achieve differential privacy on WaveCluster without requiring sophisticated sensitivity analyses of various wavelet transforms is as follows: (1) adapt an existing ϵ-differential privacy preserving data publishing method to get the noisy description of the data distribution in some fashion, such as a set of contingency tables or a spatial decomposition tree [8], [10], [18]; (2) generate a synthetic dataset according to the noisy description; (3) apply WaveCluster on the synthetic dataset. We call this technique as Baseline.

Discussion. Baseline achieves differential privacy on WaveCluster through the achievement of differential privacy on data publishing. However, it does not produce accurate

WaveCluster results. The adapted ϵ-differential privacy preserving data publishing method is designed for answering range queries. The noisy descriptions of the data distribution generated by the method may contain negative counts for certain partitions since the noise distribution is Laplacian with zero mean. These negative counts do not affect the range query accuracy since zero-mean noise distribution smooths the effect of noise. For example, a partition p_1 has the true count of 2 and the noisy count of -2, whose noise is canceled by another partition p_2 having the true count of 10 and the noisy count of 12 when both p_1 and p_2 are included in a range query. Especially when the range query spreads large range of a dataset, a single partition with noisy negative count does not affect its accuracy too much. However, when the method is used for generating a synthetic dataset, the noisy negative counts are reset as zero counts, causing the data distribution to change radically on the whole and further leading to the severe deviation in differentially private WaveCluster results.

B. Private Quantization (PrivQT)

To address the challenge faced by Baseline, we propose techniques that enforce differential privacy on the key steps of WaveCluster. Our first approach, called Private Quantization (PrivQT), introduces independent Laplacian noise in the quantization step to achieve differential privacy. In the quantization step, the data is divided into grids and the count matrix M is computed. To ensure differential privacy in this step, we rely on the Laplace mechanism that introduces independent Laplacian noise to M. Clearly, if we change one individual in the input data, such as adding, removing or modifying an individual, there is at most one change in one entry of M. According to the parallel composition property of differential privacy, the noise amount introduced to each grid is $Lap(\frac{1}{\epsilon})$, given a privacy budget ϵ. Since the following steps of WaveCluster are carried on using the differentially private count matrix M', the clusters derived from these steps are also differentially private.

Selecting the appropriate grid size (reflected by the parameter num_grid) in the quantization step strongly affects the accuracy of WaveCluster results [3], and also the differentially private WaveCluster results. A small grid size causes more data points to fall into each grid and thus the count of data points for each grid becomes larger, which makes the count matrix M resistant to Laplacian noise. However, the small grid size is not helpful for WaveCluster to detect accurate shapes of clusters and renders the results useless. On the other hand, although posing a larger grid size on the data can capture the density distribution of the data more clearly, it makes each grid's count too small and thus become sensitive to Laplacian noise, which dramatically affects the identification of significant grids and further the shapes of clusters. Our empirical experiments show that only when an appropriate grid size is given, differentially private WaveCluster results maintains high utility.

Discussion. Although PrivQT achieves differential privacy on the WaveCluster results, the noisy count matrix M' significantly distorts the noisy absolute density value d' and consequently the clustering results. The reason is as follow. Given a specified percentage value p, WaveCluster computes the absolute density value d from the positive values in $WTVM$ (shown in Figure 3(a)), while PrivQT computes d' from the positive values in $WTVM'$ (shown in Figure 3(b)). $WTVM'$ in PrivQT is derived from M', which is perturbed by Laplacian noise. Normally, Laplacian noise is non-zero, making almost all of the zero-count grids become non-zero-count grids. According to its randomness, approximately half of the zero-count grids become noisy positive-count grids due to positive noise while the remaining ones are turned into noisy negative-count grids due to negative noise. These non-zero-count grids may cause their corresponding wavelet transformed values in $WTVM'$ to become positive (depending on the targeted wavelet transform), which will inappropriately participate in the computation of d' and further distorts d'. As shown in Figure 3(b), the *red* segment moves towards the beginning of the sorted list due to those newly generated noisy positive-count grids, which later inappropriately participate in the computation of d'. Therefore, d' in PrivQT becomes much smaller than d, which causes more original non-significant grids to become significant. Figure 3(a) and (b) clearly show the comparison between d and d', and the significant grids generated by WaveCluster and PrivQT, respectively. Due to the noisy positive values in $WTVM'$, our empirical results show that the utility of differentially private WaveCluster results improves marginally even when a large privacy budget is given.

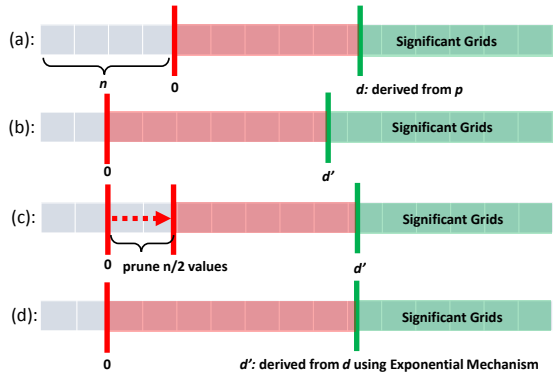


Fig. 3: Significant grid identification using a sorted list in ascending order from (a) $WTVM$; (b) $WTVM'$ in PrivQT; (c) $WTVM'$ in PrivTHR; (d) $WTVM'$ in PrivTHR$_{EM}$. *Grey* segment: zero or negative values; *Red* segment: positive values $< d$ or d'; *Green* segment: positive values $\geq d$ or d'

C. Private Quantization with Refined Noisy Density Threshold (PrivTHR)

Motivated by the limitation of PrivQT, we propose a technique, PrivTHR, which prunes a portion of noisy positive values in $WTVM'$ to refine the computation of d'. Algorithm 1 shows the pseudocode of PrivTHR.

PrivTHR first introduces random noise to the count matrix M, similar to PrivQT, and obtains a noisy count matrix M' (Line 2). PrivTHR then applies wavelet transforms on M' to obtain $WTVM'$ (Line 3). $WTVM'$ is then turned into a list

L' that keeps only positive values and the values in L' is sorted in ascending order (Line 4). Thus, only the positive values in $WTVM'$ will be used for computing d' based on the specified density threshold p. To reduce the distortion of d', starting from the smallest noisy positive values in L', PrivTHR discards the first $\frac{n'}{2}$ values (Line 6), where n is the number of zero values in the $WTVM$ and n' is a noisy estimate of n (Line 5), as shown in Figure 3(c). The reason why PrivTHR removes $\frac{n'}{2}$ values from L' is based on the insight that approximately $\frac{n}{2}$ zero values in $WTVM$ are turned into positive values due to the randomness of Laplacian noise. Since n partially describes the data distribution and releasing n without protection may leak private information, PrivTHR also introduces Laplacian noise to n, ensuring the whole process correctly enforces differentially privacy (Lines 10-16). Finally, PrivTHR applies the connected component labeling algorithm to identify clusters of significant grids (Line 8).

Algorithm 1 PrivTHR

Input: Dataset D, num_grid g, density threshold p, differential privacy budget ϵ, allocation percentage α
Output: A set of differentially private clusters (of significant grids)

1: **procedure** $PrivTHR(D, g, p, \epsilon, \alpha)$
2: M' = PrivQuantization($D, g, \alpha\epsilon$)
3: $WTVM'$ = WaveletTransform(M')
4: L' = ConvertToPosSortedArray($WTVM'$)
5: n' = NoisyCountOfZeroValues($D, g, (1-\alpha)\epsilon$)
6: L'' = RemoveFrom(L',0,$\frac{n'}{2}$)
7: d' = Percentile(L'', p)
8: **return** ConnCompLabel($WTVM', d'$)
9: **end procedure**

10: **procedure** NoisyCountOfZeroValues(D, g, ϵ)
11: M = Quantization(D, g)
12: $WTVM$ = WaveletTransform(M)
13: n = CountOfZero($WTVM$)
14: n' = n + $Lap(\frac{1}{\epsilon})$
15: **return** n'
16: **end procedure**

Budget Allocation. PrivTHR first introduces Laplacian noise in the quantization step using a privacy budget $\alpha\epsilon$, where $0 < \alpha < 1$. In the significant grid identification step, PrivTHR further introduces Laplacian noise to n using the remaining privacy budget $(1-\alpha)\epsilon$. In general only a small amount of budget is needed to obtain an estimate of n. The empirical results in Section VI will show in detail the impact of α on clustering accuracy.

D. Private Quantization with Noisy Threshold using Exponential Mechanism (PrivTHR$_{EM}$)

Besides pruning positive values in $WTVM'$, we propose an alternative technique that employs Exponential mechanism for deriving d' from the sorted list of $WTVM$. Algorithm 2 shows the pseudocode of PrivTHR$_{EM}$.

PrivTHR$_{EM}$ first introduces Laplacian noise to the count matrix M, which is similar to PrivQT and PrivTHR. After that, we obtain a noisy count matrix M' (Line 2) and the corresponding $WTVM'$ (Line 3). Different from the previous two techniques that compute d' from $WTVM'$, PrivTHR$_{EM}$ derives d' from $WTVM$ using Exponential mechanism (Lines 7-15). As demonstrated in Figure 3(d), although the sorted list derived from $WTVM'$ is severely distorted in PrivTHR$_{EM}$, the sorted list does not affect the derivation of d' at all. Given sufficient privacy budget, d' derived from Exponential mechanism is reasonably accurate, compared to the case when d' is derived from $WTVM'$.

The quality function fed into the Exponential mechanism is [8]:

$$q(L, X) = -|rank(x) - rank(d)|,$$

where L represents the sorted positive values in $WTVM$ with Min and Max values (Line 10), and X represents the possible output space, i.e., all the possible values in the range of $(0, Max]$. Given a $WTVM$ with k positive values $x_1, x_2, \ldots, x_k$, these k values divide the range $(0, Max]$ into k partitions: $(0, x_1], (x_1, x_2], \ldots, (x_{k-1}, x_k]$, and the ranks for these partitions are 1, 2, ..., k. For any $x \in (x_{i-1}, x_i]$, its rank is $rank(x_i)$. For example, if $x \in (0, Min]$, $rank(x) = rank(Min) = 1$. The sensitivity of $q(L, X)$ is 1, since any single change in the input only causes the outcome of $q(L, X)$ to be changed by 1 at maximum.

Plugging in the above quality function into Exponential mechanism, we obtain the following algorithm: for any value $x \in (0, Max]$, the Exponential mechanism (EM) returns x with probability $Pr[EM(L) = x] \propto exp(-\frac{\epsilon|rank(x)-rank(d)|}{2})$ (Line 13). Since all the values in a partition have the same probability to be chosen, a random value from the partition $Pt_i = (x_{i-1}, x_i]$ will be chosen with the probability proportional to $|Pt_i| * exp(-\frac{\epsilon}{2}|i - rank(d)|)$. In other words, once Pt_i is chosen, PrivTHR$_{EM}$ further computes a uniform random value from Pt_i as d' (Line 14).

Budget Allocation. Similar to PrivTHR, the privacy budget is split between two steps: introduction of Laplacian noise in quantization and obtaining d' using Exponential mechanism. Previous empirical experiments [8] on splitting budgets between median and count noise suggest that, 30% vs. 70% budget allocation strategy performs best. Specifically, 70% of budget is allocated for obtaining noisy count matrix M' (Line 2) and the remaining budget is allocated for computing d' (Line 4).

V. Quantitative Measures

To quantitatively assess the utility of differentially private WaveCluster, we propose two types of measures for measuring the dissimilarity between true and differentially private WaveCluster results. The first type focuses on dissimilarity of significant grids (DSG) and dissimilarity of clusters of significant grids (DSG_C). Given true and differentially private

WaveCluster results, DSG and DSG_C respectively measure the dissimilarity of the significant grids and the clusters, which are the outputs of two key steps in WaveCluster, significant grid identification and cluster identification.

The second type focuses on observing the usefulness of differentially private WaveCluster results for further data analysis. The reason is that a slight difference in the significant grids or clusters may cause a significant difference when using the WaveCluster results. In this paper, we choose a typical application of further data analysis: building a classifier from the clustering results to predict unlabeled data [28]. The classifier built from true WaveCluster results is called the true classifier clf_t while the classifier built from differentially private WaveCluster results is called the private classifier clf_p. Given the same test data, if clf_p has the same prediction result as clf_t, we regard the differentially private WaveCluster results maintain the maximum utility. To measure the dissimilarity between clf_t and clf_p, we propose two metrics: OCM and $2CE$, where OCM expresses the dissimilarity through predicted class mappings and $2CE$ expresses the dissimilarity through the relationships among the test samples with predicted classes. Depending on the targeted data analyses, either OCM or $2CE$ may provide a more reasonable estimation of the usefulness for the differentially private WaveCluster results.

A. Dissimilarity based on Significant Grids and Clusters

This section presents the first type of dissimilarity measures: DSG and DSG_C. We use T to denote the set of significant grids in the true WaveCluster results and P to denote the set of significant grids in the differentially private WaveCluster results.

Dissimilarity of Significant Grids (DSG). DSG captures the differences regarding the significant grids. For example, if a grid a is significant in the true WaveCluster results but insignificant in the differentially private WaveCluster results, we reflect this difference in DSG. DSG is formally defined as follows:

$$DSG = \frac{|T \bigcup P - T \bigcap P|}{|T|}$$

DSG computes the ratio of the count of dissimilar significant grids between T and P to the count of significant grids in T. The smaller the DSG is, the higher utility the private WaveCluster results maitain. As an extreme, when T is exactly the same with P, DSG returns the minimum. For another extreme, when T is totally different from P, DSG could be exceptionally large.

Dissimilarity of Clusters of Significant Grids (DSG_C). DSG reflects the dissimilarity on only significant grids without measuring how far apart the clusters are in the true and differentially private WaveCluster results. Thus, we further propose DSG_C, which considers the dissimilarities of both significant grids and clusters. Assume that there are t clusters of true significant grids and s clusters of differentially private significant grids. t might not be equal to s, and the cluster labels in t true clusters and s private clusters are completely arbitrary. To accommodate these differences, we adopt the Hungarian method [29], [30], a combinatorial optimization algorithm, to solve the matching problem between t true clusters and s private clusters while minimizing the matching difference.

Algorithm 2 PrivTHR$_{EM}$

Input: Dataset D, num_grid g, density threshold p, differential privacy budget ϵ, allocation percentage α
Output: A set of differentially private clusters (of significant grids)

1: **procedure** $PrivTHR_{EM}(D, g, p, \epsilon, \alpha)$
2: M' = PrivQuantization($D, g, \alpha\epsilon$)
3: $WTVM'$ = WaveletTransform(M')
4: d' = NoisyDensityThresholdEM($D, g, p, (1-\alpha)\epsilon$)
5: **return** ConnCompLabel($WTVM', d'$)
6: **end procedure**

7: **procedure** NoisyDensityThresholdEM(D, g, p, ϵ)
8: M = Quantization(D, g)
9: $WTVM$ = WaveletTransform(M)
10: L = ConvertToPosSortedArray($WTVM$)
11: d = Percentile(L, p)
12: $rank_d = \lceil \frac{p}{100} * length(L) \rceil$
13: $rank_{d'}$ = ExponentialMechanism(L,$rank_d$,ϵ)
14: d' = UniformRandom(L, $rank_{d'} - 1$,$rank_{d'}$)
15: **end procedure**

When cluster C_i matches to cluster C_j, we define that the distance d between cluster C_i and cluster C_j is $max\{|C_i \backslash C_j|, |C_j \backslash C_i|\}$. Consider a cluster $C_i = \{g_1, g_3, g_5\}$ and a cluster $C_j = \{g_1, g_5, g_7, g_9\}$. The distance d between clusters C_i and C_j is $max\{|\{g_3\}|, |\{g_7, g_9\}|\} = 2$. Given t true clusters and s private clusters, assuming that $t \geq s$, a matching $M_{t,s}$ of t true clusters and s private clusters is a set of cluster pairs, where each private cluster is matched with a true cluster. We then define the cost of a matching (M_{cost}) as the sum of all the distances between each cluster pair in the matching $M_{t,s}$ plus the count of significant grids in the non-matched clusters:

$$M_{cost} = \sum_{1 \leq i_x \leq t, 1 \leq j_y \leq s} max\{|C_{i_x} \backslash C_{j_y}|, |C_{j_y} \backslash C_{i_x}|\} + \sum_{1 \leq z \leq t} |C_z|$$

Here, i_x and j_y indicate the subscripts of clusters in a matched pair. $|C_z|$ represents the count of significant grids in the non-matched true clusters. Among all the possible matchings of clusters, we use the Hungarian method to find the optimal matching with the minimum M_{cost}. Based on the minimum M_{cost}, DSG_C is computed as follows:

$$DSG_C = \frac{M_{cost}}{|T|}$$

For one extreme, the Hungarian method finds the matching with M_{cost} being 0, when t true clusters and s private clusters

only differ in cluster labels.

B. Dissimilarity based on Classifier Prediction

This section presents the second type of dissimilarity measures, OCM and $2CE$, to measure the dissimilarity between clf_t and clf_p. We name this way of evaluation as "clustering-first-then-classification": given a set of unlabeled data points, we use a portion of the data points (e.g., 90%) to compute WaveCluster results, where each cluster is a set of significant grids. Using the significant grids with cluster labels as training data, we build classifiers clf_t and clf_p, and use them to predict the classes for the remaining data points (e.g., 10%).

Dissimilarity of Classifiers based on Optimal Cluster Matching (OCM). OCM measures the dissimilarity between the two sets of classes predicted by clf_t and clf_p for the same test samples. We use L_t to denote the set of classes predicted by clf_t and L_p to denote the set of classes predicted by clf_p. Since L_t and L_p are completely arbitrary, we exploit the Hungarian method to find the optimal matching between L_t and L_p, which is similar to the case described in DSG_C.

Assume that a class $L_{t,i}$ predicted by clf_t is matched to a class $L_{p,j}$ predicted by clf_p, forming a class pair. We compute the count of common test samples in the class $L_{t,i}$ and the class $L_{p,j}$, and sum all the common test samples in each class pair to compute CT:

$$CT = \sum_{1 \le i \le c_1, 1 \le j \le c_2} |L_{t,i} \bigcap L_{p,j}|$$

Here c_1 is the count of classes in L_t and c_2 is the count of classes in L_p, and we assume $c_1 \ge c_2$. Since there are many possible mappings from the classes in L_t to the classes in L_p, we use the Hungarian method to find the optimal mapping that maximizes CT. Based on CT and the total count of the test samples TT, we derive the dissimilarity OCM:

$$OCM = 1 - \frac{CT}{TT}$$

When the dissimilarity is smaller, the differentially private WaveCluster results are more similar to the true WaveCluster results and maintain high utility for classification use.

Dissimilarity of Classifiers based on 2-Combination Enumeration ($2CE$). $2CE$ measures the dissimilarity between clf_t and clf_p based on relationships of every pair of test samples, i.e., whether two samples are in the same cluster. Essentially, given a pair of test samples A and B, we say A and B are classified consistently either (1) $clf_t(A) = clf_t(B)$ and $clf_p(A) = clf_p(B)$ or (2) $clf_t(A) \neq clf_t(B)$ and $clf_p(A) \neq clf_p(B)$. $2CE$ is the ratio of the count of test sample pairs that are not classified consistently over the total number of test sample pairs, which is the set of 2-combination of the test samples. We denote the set of 2-combination as $ComList$. Consider the set of test samples $\{A, B, C, D\}$. Assume that the predicted classes from clf_t are $\{1, 2, 1, 3\}$ whereas the predicted classes from clf_p are $\{1, 2, 2, 3\}$. The values that indicate whether a sample pair is classified in the same class using clf_t and clf_p are shown in table I, where

TABLE I: $ComList$ values under clf_t and clf_p

$ComList$	AB	AC	AD	BC	BD	CD
value under clf_t	0	1	0	0	0	0
value under clf_p	0	0	0	1	0	0

0 means the sample pair is in different classes and 1 means the sample pair is in the same class. We then perform parity check on the two values of $ComList$ and count the test sample pairs with inconsistent bit values, which is 2. Therefore, $2CE$ is 2/6 for this example. When $2CE$ is 0, it means that there are no differences between clf_t and clf_p. In other words, the differentially private WaveCluster results are as useful in classification as the true ones. $2CE$ uses $ComList$ to eliminate the need of finding the optimal matching between the classes predicted by clf_t and clf_p, since $ComList$ considers whether two test samples are predicted to be in the same class. We next analyze how the value of $2CE$ reflect the dissimilarity between the predicted classes using clf_t and clf_p.

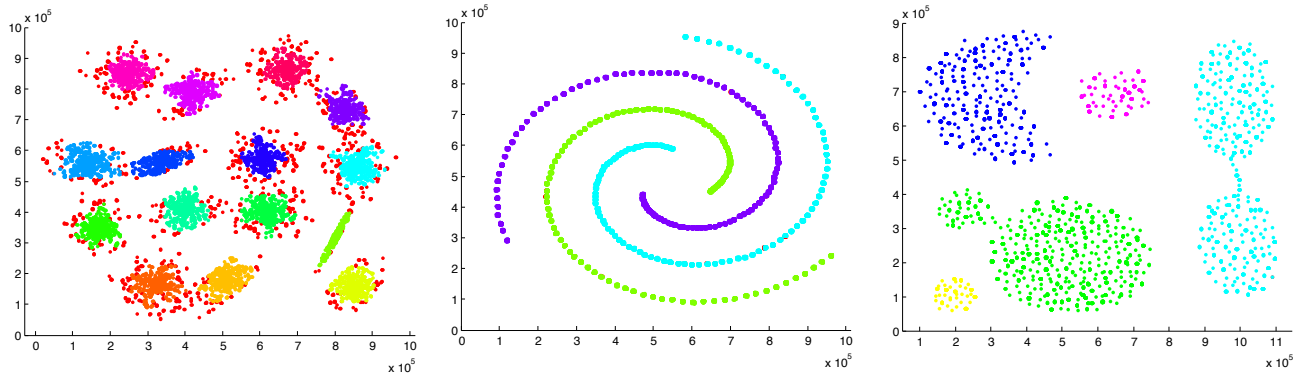
Unlike OCM which captures the class mappings, $2CE$ captures the relationships among the test samples and may have a very different value as OCM. Assume that there are N test samples and k ($0 < k \le N$) test samples have discrepancy in the predicted classes using clf_t and clf_p, which causes the values of $ComList$ to be different (e.g., the difference between the values of $ComList$ in table I originates from the prediction discrepancy for test sample C). To make the analysis easier, we also assume that these k test samples have the same predicted class using clf_t. The minimum of $2CE$ is 0 and $2CE$ can be expressed as follows:

$$2CE \le \begin{cases} \frac{C(k,2)+k(N-k)+C(N-k,2)}{C(N,2)}, & \text{if } N \ge 4 \text{ and } k \ge 2 \\ \frac{k(N-k)+C(N-k,2)}{C(N,2)}, & \text{if } 0 < N \le 3 \text{ and } k = 1 \end{cases}$$

$C(k, 2)$ represents the 2-combinations of test samples in the k test samples, $k(N-k)$ represents the 2-combinations of the k test sample and the other $N-k$ test samples, and $C(N-k, 2)$ represents the 2-combinations of test samples in the $N-k$ test samples. Since $C(N-k, 2)$ have the same predicted classes using clf_t and clf_p (based on our assumption), $C(N-k, 2)$ is always 0. When every test sample in the k test samples is predicted to be in a different class using clf_p, $2CE$ achieves the maximal value. On the contrary, when all the k test samples are predicted to be in the same class using clf_p, $C(k, 2)$ is 0 and $2CE$ equals to $\frac{k(N-k)}{C(N,2)}$. As $C(N, 2)$ grows much faster than $k(N-k)$, when N is large, $2CE$ has a low value, while OCM has a relatively large value since OCM expresses such differences as $\frac{k}{N}$. The above analysis will be used to explain some of the observations from the experiments in Section VI.

VI. Experiments

In this section, we evaluate the proposed techniques using three synthetic clustering datasets from [31]. The implementation of non-private WaveCluster algorithm is provided by [32]. In our experiments, we use Haar transform as the wavelet transform for the four techniques. The classification algorithm used for measuring OCM and $2CE$ is C4.5 [33], [34]. All



(a) *DS1*, g = 64, p = 58 (b) *DS2*, g = 40, p = 10 (c) *DS3*, g = 50, p = 15

Fig. 4: Illustration of datasets and their WaveCluster results

experiments were conducted on a machine with Intel 2.67GHz CPU and 4GB RAM.

A. Synthetic Datasets

The three synthetic clustering datasets contain different data shapes that are specially interesting in the context of clustering. We enlarge these synthetic datasets and illustrate these datasets by plotting the data points directly in Figure 4.

$DS1$: two-dimensional data containing 15 Gaussian clusters with different degrees of cluster overlapping. It contains 30000 data points. These 15 clusters are all in convex shapes. The center area of each cluster has higher density and is resistant to noise. However, the overlapped area of two adjacent clusters has smaller density and is prone to be affected by noise, which might turn the corresponding non-significant grids into significant grids and further connect two separate clusters.

$DS2$: two-dimensional spiral data with 3 clusters. It contains 31200 data points. The head of each spiral is quite close to one another. Some noisy significant grids are very likely to bridge the gap between adjacent spirals and merge them into one cluster.

$DS3$: two-dimensional data with 5 various shapes of clusters, including concave shapes. It contains 31520 data points. There are two clusters that both contain two sub components and a narrow line-shape area that bridges those two sub components. The narrow bridging area has low density and might be turned into non-significant grids, causing a cluster to split into two clusters.

Figures 4 also shows the WaveCluster results on these three datasets under certain parameter settings (i.e., the values of g and p are specified in Figure 4). Any two adjacent clusters are marked with different colors. The points in red color are identified as noise, which fall in the non-significant grids.

In our experiments, we compare the performances of the four techniques (Baseline, PrivQT, PrivTHR, and PrivTHR$_{EM}$) on these three datasets using the four metrics proposed in Section V and provide analysis on the results.

B. Dissimilarity based on Significant Grids and Clusters

We here present the results of the two dissimilarity metrics: DSG and DSG_C.

Results of DSG Figures 5 (a), (e), (i) show the results of DSG for the four techniques when the privacy budget ϵ ranges from 0.1 to 2.0. The X-axis shows the privacy budgets ϵ, and the Y-axis denotes the DSG values. As shown in the results, PrivTHR outperform Baseline and PrivQT on all datasets for all privacy budgets. PrivTHR$_{EM}$ has a similar performance as PrivTHR with sufficient privacy budgets ($\epsilon \geq 0.5$). When ϵ is very limited (i.e., 0.1), PrivTHR$_{EM}$ performs the worst among all on $DS1$. The major reason is that limited budget causes Exponential mechanism ($exp(\frac{\epsilon q}{2})$) to be insensitive for the changes of q, where q is the value of the quality function ($q = -|rank(x) - rank(d)$). Therefore, the difference between the ranks of d' and d may be large and causes DSG to increase significantly.

As ϵ increases, Baseline, PrivTHR, and PrivTHR$_{EM}$ generally achieve smaller values of DSG that indicate better utility. Larger privacy budget allows the synthetic data generation method to capture more accurate data distribution and thus Baseline performs better in identifying significant grids, which results in better DSG values. Similarly, given larger privacy budgets, both PrivTHR and PrivTHR$_{EM}$ can generate more accurate d' and further identify more similar significant grids as the non-private WaveCluster algorithm.

Unlike the other techniques, PrivQT surprisingly benefits little from increased privacy budgets. Increasing privacy budgets can only reduce noise magnitude. However, $WTVM$ can be affected by any level of noise magnitude and thus approximately half of the zero values in $WTVM$ are turned into noisy positive values in $WTVM'$, causing d' to be significantly distorted with any level of privacy budget.

We also observe that the results of DSG on these three datasets are quite different, which originates from the different data distributions of these three datasets. The 15 clusters in $DS1$ scatter quite evenly around the whole space and the shapes of all clusters are convex. Most of the significant grids

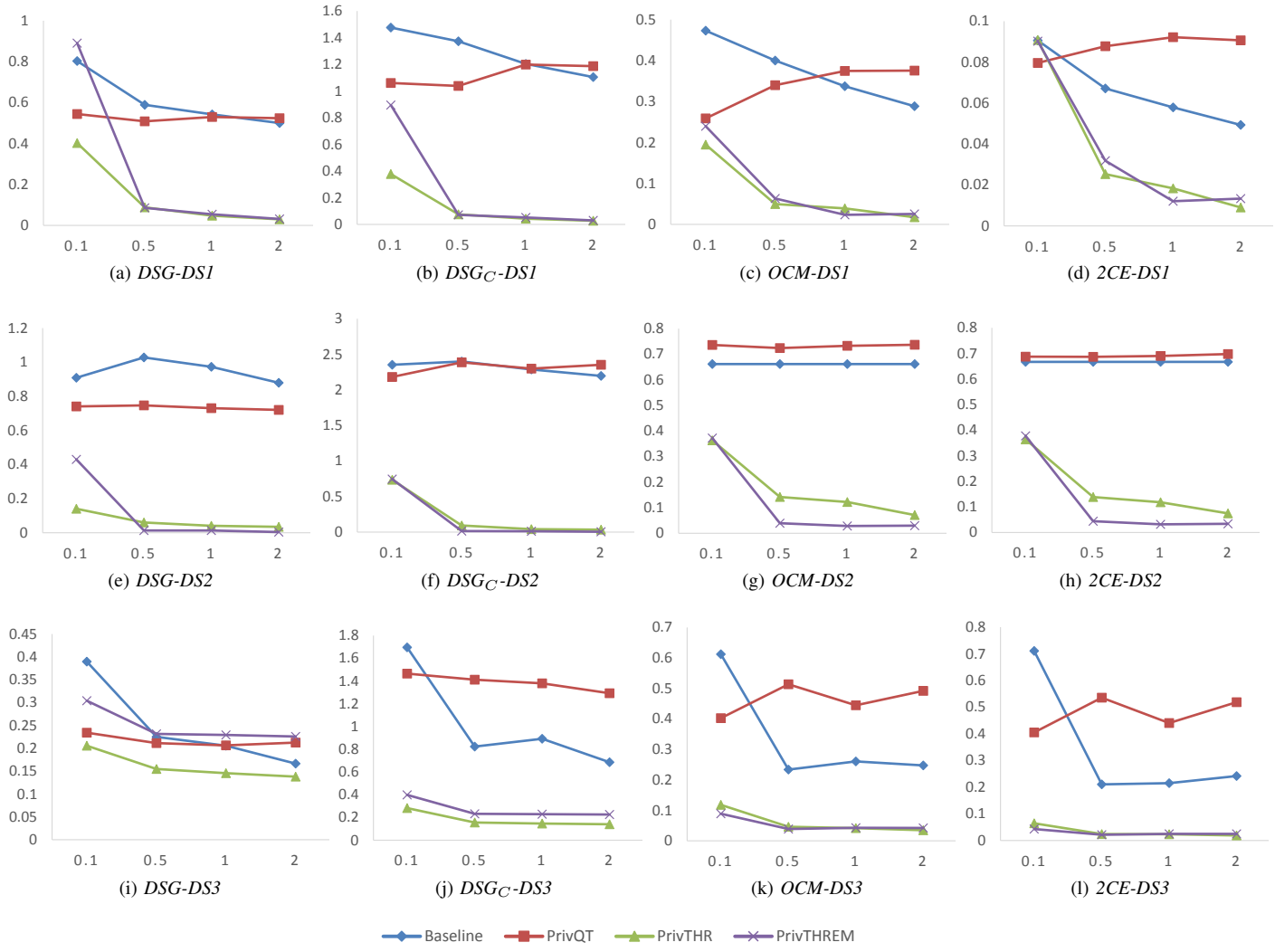


Fig. 5: Results of DSG and DSG_C on $DS1$, $DS2$ and $DS3$

located in the center of the cluster still remain significant under differential privacy and are resistant to noise. Only those grids around the border of each cluster change from significant to non-significant or from non-significant to significant, which causes the count of dissimilar grids between T and P to increase and further increases DSG. The 3 spirals in $DS2$ are in concave shapes and synthetic data generation method cannot preserve the original data distribution very accurately. The reason is that uniform random generating synthetic points in a partition gradually expands the spirals toward convex shapes. What is more, the total count of true significant grids $|T|$ are relatively small compared to the total count of grids, which makes the ratio of the dissimilar significant grids more prominent in this dataset. This explains why Baseline on $DS2$ has higher DSG values (ranging from 0.88 to 1.03) than those on $DS1$ and $DS3$ (ranging from 0.15 to 0.80). On $DS3$, all four techniques achieve similar DSG values as privacy budget increases. The reason is that the clusters in $DS3$ scatter around the whole space and are well separated without overlapping. Although $DS3$ contains two narrow line-shape areas that are sensitive to noise due to their low densities, DSG measures only the dissimilarities of significant grids and cannot capture the dissimilarities of low-density areas precisely. We can see the differences among these four techniques on $DS3$ when we use DSG_C to measure dissimilarities.

Results of DSG_C. Figures 5 (b), (f), and (j) show the results of DSG_C for the four techniques when the privacy budget ranges from 0.1 to 2.0. X-axis shows the privacy budgets, and Y-axis denotes the values of DSG_C. We found that in quite a lot of cases, DSG_C is larger than DSG when the setting is exactly the same. The reason is that DSG_C might double-count the dissimilar significant grids between T and P. For example, suppose there are two clusters in T, cluster C_1 includes significant grids $\{g_1, g_2\}$ and cluster C_2 contains grid $\{g_3\}$. In P, cluster C_1 includes significant grid $\{g_1\}$ while cluster C_2 contains grids $\{g_2, g_3\}$. According to the definition of DSG_C, we found g_2 is counted twice in DSG_C. This double-counting causes higher DSG_C value when the setting is the same.

As shown in the results, both PrivTHR and PrivTHR$_{EM}$

achieve smaller DSG_C values than Baseline and PrivQT on all three datasets for all budgets. The reason is that DSG_C considers the differences of both significant grids and clusters. Therefore even if the noisy significant grids are similar to the true significant grids, these noisy significant grids may result in very different shapes of clusters and thus result in a large value of DSG_C. For example, on $DS1$ with ϵ being 0.1, PrivTHR$_{EM}$ has the worst DSG value among the four techniques since Exponential mechanism cannot produce a very precise estimation of d'. However, its DSG_C value is smaller than Baseline and PrivQT, since the noisy clusters formed by the noisy significant grids of Baseline and PrivQT are quite different from the true clusters in terms of number of clusters and data shapes. Similarly, although four techniques achieve similar DSG values on $DS3$, their DSG_C values are quite different. In $DS3$, the narrow line-shape areas and the gap between two adjacent clusters are sensitive to noise. If some noisy significant grids appear in these areas, two clusters may be merged into one; if some significant grids disappear due to noise, one cluster might be split into two clusters. Such changes cause DSG_C to increase significantly since DSG_C measures the differences of clusters rather than the differences of significant grids.

Similar to the results of DSG, Baseline, PrivTHR, and PrivTHR$_{EM}$ generally achieve smaller values of DSG_C as ϵ increases. PrivQT still benefits little from the increases of ϵ due to the noisy positive values in $WTVM'$ as described in the analysis of DSG.

C. Dissimilarity based on Classifier Prediction

We here present the results of the two dissimilarity metrics: OCM and $2CE$.

Results of OCM. Figures 5 (c), (g), and (k) show the results of OCM for the four techniques when the privacy budget ϵ ranges from 0.1 to 2.0. X-axis denotes the privacy budgets while Y-axis denotes the values of OCM. As shown in the results, PrivTHR and PrivTHR$_{EM}$ achieve smaller OCM values than Baseline and PrivQT for all datasets when ϵ ranges from 0.1 to 2.0. When ϵ is greater than 0.5, the OCM values of PrivTHR and PrivTHR$_{EM}$ are less than 0.1 on $DS1$ and $DS3$, indicating the private classifier clf_p maintains highly similar prediction results as the true classifier clf_t. On $DS2$ that contains 3 spirals, PrivTHR$_{EM}$ still maintains a very low OCM value (< 0.1) when ϵ is greater than 0.5 while PrivTHR has a slightly worse OCM value (ranging from 0.1 to 0.2). Such results show that PrivTHR$_{EM}$ is more resilient to noise for concave-shaped data than PrivTHR.

The different data distributions of the three datasets cause Baseline and PrivQT to have very different values of OCM. On $DS1$, although Baseline does not perform better than PrivTHR and PrivTHR$_{EM}$, Baseline exhibits a steadily decreasing trend of OCM values as ϵ increases. The reason is that all 15 clusters in $DS1$ are convex shaped, which can be well captured by synthetic data generator when a larger ϵ is given. On $DS2$ and $DS3$, both Baseline and PrivQT achieve larger OCM values. $DS2$ contains 3 spiral-shaped clusters that can be easily merged into one cluster by noisy significant grids. For Baseline, resetting negative counts to zero counts makes some noisy positive-count grids too prominent and become one of the "bridging points" that wrongly connect the clusters. For PrivQT, approximately half of the positive values in $WTVM'$ cause certain grids to become these "bridging point". This explains why Baseline and PrivQT both perform steadily worse and have an OCM value more than 0.66 on $DS2$. Similarly, $DS3$ has several unique shapes, and two adjacent clusters are easily merged into one when noisy significant grids show up in the gap between two clusters. As we can see from the results, while OCM of Baseline decreases when ϵ is 0.5 on $DS3$, such decreasing trend does not continue when $\epsilon > 0.5$. In fact, the occurrence of noisy bridging significant grids is quite random, and thus PrivQT and Baseline do not exhibit an obvious increasing or decreasing trend of OCM when the budget increases on $DS3$.

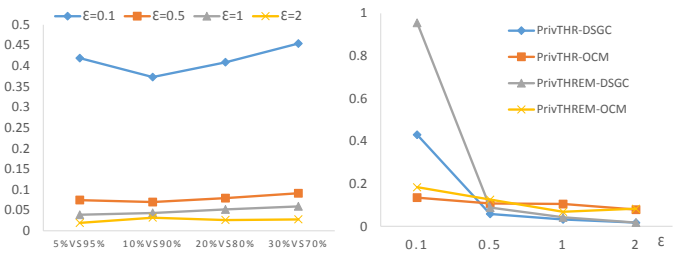
Similar to the results of DSG and DSG_C, Baseline, PrivTHR, and PrivTHR$_{EM}$ generally achieve smaller values of OCM as ϵ increases. The results also reconfirm that PrivQT benefits little from the increases of ϵ due to the noisy positive values in $WTVM'$.

Results of $2CE$. Figures 5 (d), (h), and (l) show the results of $2CE$ for the four techniques when the privacy budget ϵ ranges from 0.1 to 2.0. X-axis denotes the privacy budgets while Y-axis denotes the values of $2CE$. As shown in the results, PrivTHR and PrivTHR$_{EM}$ achieve smaller OCM values than Baseline and PrivQT for all datasets when ϵ ranges from 0.1 to 2.0.

In general, all four techniques exhibit similar trends of $2CE$ as their trends in OCM. On $DS1$, all four techniques have very low $2CE$ values (< 0.1) though their corresponding OCM values are much higher (ranging from 0.05 to 0.5). The reason is that $2CE$ captures the relationships between data points while OCM focuses on the mappings of classes. As described in Section V-B, if there are k test samples out of N total samples have different prediction results in the true and private results, $2CE$ expresses the differences as $C(k,2)+k(N-k)$ over the total combinations of test samples $C(N,2)$, while OCM expresses the differences as k over N. On $DS1$, the k test samples are predicted to be in the same cluster in the private results and $C(k,2)$ becomes close to 0. In this case, only $k(N-k)$ matters in the computation of $2CE$. Given that $C(N,2)$ is much larger than N and $k(N-k)$ when N of $DS1$ is about 30,000, $2CE$ has a smaller value than OCM for measuring the differences, and thus is less sensitive to the noise on $DS1$.

Similar to the results of other dissimilarity measures, Baseline, PrivTHR, and PrivTHR$_{EM}$ generally achieve smaller values of $2CE$ as ϵ increases. PrivQT exhibits a random trend of $2CE$ as ϵ increases, and still benefits little from the increases of ϵ.

Budget Allocation for PrivTHR. Figure 6 (a) shows the values of DSG for PrivTHR under different budget allocation strategies. As we can see from the results, the budget allocation strategy with 10% for n (the number of positive values in



(a) (b)

Fig. 6: (a) PrivTHR with different budget allocation strategies using *DSG*. (b) PrivTHR and PrivTHR$_{EM}$ using Biorthogonal wavelet transform on *DS1*

$WTVM$) performs the best, i.e., 90% of budget is used for private quantization while 10% of budget is used for obtaining n'. Due to space limit, we show only DSG values on $DS1$. The results of other measures on $DS1$ show the similar results, and the results of all the four measures on $DS2$ and $DS3$ also show the similar results.

Different Wavelet Transform. Figure 6 (b) shows the results of DSG_C and OCM for PrivTHR and PrivTHR$_{EM}$ using Biorthogonal transform (bior2.2) on $DS1$. As we can see from the results, both PrivTHR and PrivTHR$_{EM}$ exhibit decreasing trends of DSG_C and OCM, similar to the trends shown in Figures 5 (b) and (c), which use Haar transform. Such results demonstrate that our techniques are independent of the used wavelet transform. Due to the space limit, here we provide the results of only $DS1$ and use DSG_C and OCM as measures. The results of all the measures on $DS2$ and $DS3$ show the similar trends.

VII. Conclusion

In this paper we have addressed the problem of cluster analysis with differential privacy. We take a well-known effective and efficient cluster-analysis algorithm called WaveCluster, and propose several ways to introduce randomness in the computation of WaveCluster. We also devise several new quantitative measures for examining the dissimilarity between the non-private and differentially private results and the usefulness of results in classification. In the future, we will investigate under differential privacy other categories of cluster-analysis algorithms, such as hierarchical clustering. Another important problem is to explore the applicability of differentially private cluster analysis in those cases where the users do not have good knowledge about the dataset, and the parameters of the algorithms should be inferred in a differentially private way.

References

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[2] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases," in *VLDB*, 1998, pp. 428–439.

[3] ——, "Wavecluster: A wavelet-based clustering approach for spatial data in very large databases," *The VLDB Journal*, vol. 8, no. 3-4, pp. 289–304, 2000.

[4] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*. ISBAN 012047140X, Orlando, FL, USA: Academic Press, Inc., 1992.

[5] S. G. Mallat, *A Wavelet Tour of Signal Processing. Academic Press*. ISBN 978-0-12-466606-1: Academic Press, Inc., 1999.

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

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

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

[9] W. H. Qardaji, W. Yang, and N. Li, "Differentially private grids for geospatial data," in *ICDE*, 2013, pp. 757–768.

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

[11] B. K. P. Horn, "Robot vision," in *The MIT Press, forth edition*, 1988.

[12] C. Dwork, "Differential privacy: A survey of results," in *TAMC*, 2008, pp. 1–19.

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

[14] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *STOC*, 2009, pp. 371–380.

[15] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim, "Private coresets," in *STOC*, 2009, pp. 361–370.

[16] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy, accuracy, and consistency too: A holistic solution to contingency table release," pp. 273–282, 2007.

[17] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 1021–1032, 2010.

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

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

[20] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proc. VLDB Endow.*, vol. 5, no. 11, pp. 1364–1375, 2012.

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

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

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

[24] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," *Commun. ACM*, vol. 53, no. 9, pp. 89–97, 2010.

[25] M. Winslett, Y. Yang, and Z. Zhang, "Demonstration of damson: Differential privacy for analysis of large data," *ICPADS, IEEE Computer Society*, pp. 840–844, 2012.

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

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

[28] P. Green, F. J. Carmone, and S. M. Smith, "Multidimensional scaling, section five: Dimension reducing methods and cluster analysis," 1989, addison Wesley.

[29] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[30] ——, "Variants of the hungarian method for assignment problems," *Naval Research Logistics Quarterly*, vol. 3, p. 253258, 1956.

[31] "Clustering datasets," http://cs.joensuu.fi/sipu/datasets/.

[32] "Wavecluster," http://michael.barnathan.name/v2/cgi-bin/listdocuments.cgi?category=2.

[33] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufmann Publishers Inc., 2011.

[34] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.