Journal of Communications and Information Networks, Vol.2, No.3, Sept. 2017
DOI: 10.1007/s41650-017-0014-x

Research paper

Different strategies for differentially private histogram publication

Xue Meng[1], Hui Li[1*], Jiangtao Cui[2]

1. School of Cyber Engineering, Xidian University, Xi'an 710071, China
2. School of Computer Science and Technology, Xidian University, Xi'an 710071, China

* Corresponding author, Email: hli@xidian.edu.cn

Abstract: Differential privacy is a strong notion for protecting individual privacy in data analysis or publication, with strong privacy guaranteeing security against adversaries with arbitrary background knowledge. A histogram is a representative and popular tool for data publication and visualization tasks. Following the emergence and development of data analysis and increasing release demands, protecting the private data and preventing sensitive information from leakage has become one of the major challenges for histogram publication. In recent years, many approaches have been proposed for publishing histograms with differential privacy. This paper explores the problem of publishing histograms with differential privacy, and provides a systematical summarization of existing research efforts in this field, begining with a discussion of the basic principles and characteristics of the technology. Furthermore, we provide a comprehensive comparison of a series of state-of-the-art histogram publication schemes. Finally, we provide possible suggestions for further expansions of future work in this area.

Keywords: differential privacy, histogram, Laplacian noise, data publication, accuracy, utility

Citation: X. Meng, H. Li , J. T. Cui. Different strategies for differentially private histogram publication [J]. Journal of communications and information networks, 2017, 2(3): 68-77.

1 Introduction

A histogram is a summary of the occurrence counts for the values in a particular domain in a given dataset. Histograms are an effective and popular statistical tool for various applications, such as linear range queries in Refs. [1,2], as well as data mining and analysis[3]. To facilitate these applications, histograms are generated from a raw dataset and published for answering analytical queries. Tab. 1 is an example of a raw data table that stores the information of some patients in a particular hospital. A common task in disease analysis and diagnosis studies involves requesting for range queries or summarizations of diseases based on some particular patient features. To answer range queries and facilitate analytical tasks, histograms are pre-computed based on raw data (Tab. 1), resulting in a graph like the one shown in Fig. 1. As shown in Fig. 1, a histogram is in fact displays a group of bins, each of which corresponds to a particular disease and is associated with a height as the cardinality of patients belonging to that group.

With the emergence of the Internet of Things, Cloud computing, and mobile techniques, it is now relatively easy for various organizations to easily gather

Manuscript received Sept. 23, 2016; accepted Dec. 21, 2016
This work is supported by the National Nature Science Foundation of China (Nos. 61672408, 61472298), the National High Technology Research and Development Program ("863" Program) (No. 2015AA016007), the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2015JQ6227) and China 111 Project (No. B16037).

vast amounts of personal information, such as medical records, web search history, and traffic monitoring records, etc. A hospital gathers data from individual parents every day. These dynamic datasets (daily datasets of individual patients with fevers, coughs, and different demographic attributes) can be shared between researchers to aid in cohort discovery. A GPS service provider gathers data from individual users including their location, speed, mobility, etc. The dynamic datasets, e.g., the numbers of users in different regions during various time periods, can be mined from this raw data for commercial interests such as determining congestion patterns on the roads.

Table 1 Raw patient table

name	age	disease
Mike	30	flu
Casse	35	fever
Bob	26	HIV
Amy	41	flu
Tod	50	HIV
Ann	33	flu
...	...	...

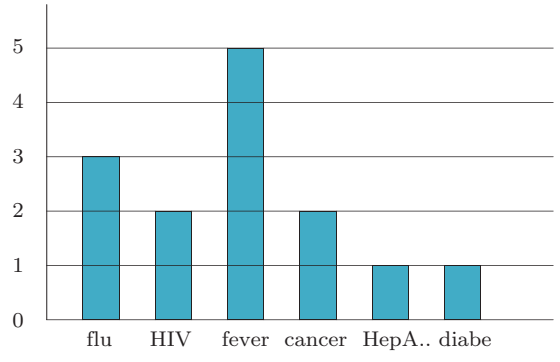

Figure 1 Histogram of patient table

Analysis of such data can yield valuable insights, including new understanding of a disease, or typical consumer behavior in a community. However currently privacy concerns have become a major hurdle for such analysis for two aspects. First, it has been increasingly difficult for third-party data analysts to access their input data. For example, medical researchers have to routinely obtain the approval of their respective institutional review boards, which is tedious and time-consuming, before they are able to look at the data they need. Second, privacy concerns complicate the publication of results.

The publication of histograms carries the risk of leaking sensitive information about the underlying dataset. Therefore, the publication of histograms must consider any privacy issues associated with the data. For this reason, many research efforts have been made to facilitate privacy-preserving histogram publication.

In recent years, significant breakthroughs have been made in differential private histogram publication. The ultimate goal is to ensure that range queries on a private histogram can be answered as accurately as possible in a manner that preserves privacy.

Among the existing privacy-preserving data publication schemes, including anonymization, sanitization, etc., differential privacy is the most promising technique as it provides a theoretical guarantee of privacy, for the published data, against any adversary with background knowledge. In particular, through careful investigation, we have classified the existing differential privacy histogram publication schemes into two groups.

In the first group, coarse histogram bins are sanitized using some carefully designed mechanisms, e.g., Laplacian Mechanism, Exponential Mechanism, etc. After which, the sanitized histograms are then post-processed using particular optimization strategies, such that the error introduced by the sanitization is minimized as much as possible, assuring usability of the histogram data. The second group works in the opposite direction. Coarse histogram bins are first preprocessed using particular transformations or optimizations. The transformed data are then injected with Laplacian or Exponential noises in order to ensure that the output histogram is Differential Private-Compliant. Both the groups of methods aim to provide a theoretical guarantee of privacy for the published histogram as much as possible while preserving the usability of the published data.

We describe a series of preliminaries in section 2.

We then present a classification of existing differential private histogram publication mechanisms and compare them in section 3. We describe our conclusions in section 4. Triggered expansion and future work is then discussed in section 5.

2 Preliminaries

2.1 Histogram publication

A histogram is a common representation of a distribution of numerical or categorical data. Given a series of n data samples $D = \{\mathbb{X}_1, \mathbb{X}_2, \cdots, \mathbb{X}_n\}$, the value of each record x_i in a particular field (resp., attribute), say A, is denoted $A(x_i)$, A histogram of field A in D merges neighboring counts into k equal-width bins $H = \{\mathbb{B}_1, \mathbb{B}_2, \cdots, \mathbb{B}_k\}$, each of which is disjoint with the others. Each bin holds a count of the records that fall into it. In the case where $A(x_i)$ are numerical values, each bin $B_j = (l_j, r_j, c_j)$ contains an interval $[l_j, r_j] \subseteq [1, n]$, and a count c_j which represents the number of records in D that fall within the interval $[l_j, r_j]$, i.e. $\{x_i | l_j \leqslant i \leqslant r_j\}$. The bins in a histogram must be disjoint but collectively cover all the counts in D.

A coarse histogram can be further summarized and approximated by a higher level histogram H, that may contain fewer bins, each of which covers a larger interval. Because such an abstracted histogram uses fewer counts than the original coarse histogram D, it inevitably introduces error. This error is often measured using an SSE (Sum of Squared Errors)[4] between the raw histogram D and the generalized histogram H as follows

$$SSE(H, D) = \sum_j \sum_{l_j \leqslant i \leqslant r_j} (c_j - c_i)^2. \tag{1}$$

$SSE(D, H)$ can be used to evaluate the usability of the histogram, For the interval $[l_j, r_j]$ of each bin B_j, the optimal value of c_j for B_j that minimizes $SSE(H, D)$ is simply the mean value of the counts in $[l_j, r_j]$, i.e. $j = (\sum_{i=l_j}^{r_j} x_i)/(r_j - l_j + 1)$.

2.2 DP (Differential Privacy)

Given a set of records $D = \{\mathbb{X}_1, \mathbb{X}_2, \cdots, \mathbb{X}_n\}$, another record set D' is a neighboring sequence to D, if and only if any D' differs from D in only one record. meaning D' can be generated from D by either inserting or removing a single record.

A histogram publication mechanism Q satisfies ϵ-differential privacy (ϵ-DP)[5], if it outputs a randomized histogram H, such that $\forall D, D', Pr(Q(D) = H) \leqslant e^z \times Pr(Q(D') = H)$, where D and D' denote two arbitrary neighboring datasets, and $Pr(Q(D) = H)$ denotes the probability that Q outputs H with input D. From the definition, it can be seen that the differential privacy model distorts the effect of some data records in the algorithm output, such that the records from query Q are indistinguishable to arbitrary adversaries.

The first and most commonly used mechanism for achieving differential privacy is the Laplacian mechanism, which relies on Laplacian Distributions. To achieve ϵ-differential privacy with the Laplacian Mechanism, we must first discuss the concept of sensitivity. The sensitivity Δ[6] of the query (e.g., a histogram query in our problem) is defined as the maximum L_1-distance between the exact answers of the query Q on any two neighboring databases D and D', i.e.,

$$\Delta Q = \max_{D, D'} \|Q(D) - Q(D')\|_1. \tag{2}$$

2.3 LM (Laplacian Mechanism)

A standard mechanism for achieving differential privacy is to add Laplacian noise to the original output of a function f. This means that the noise added to the original outputs is drawn from a Laplacian distribution. This Laplacian Mechanism was proposed by Dwork et al[7]. Formally, it takes as input a database D, a function f, and the privacy parameter ϵ as inputs. The noise is generated according to a Laplacian distribution with the probability density function $p(x|\lambda) = \frac{1}{2\lambda} \exp(-|x|/\lambda)$, where λ is determined by both Δf and the desired privacy parameter ϵ. Then, the Laplacian mechanism works as follows.

Theorem 1 For any function $f : D \rightarrow R^d$, the mechanism A, which is defined as

$$A(D) = f(D) + \langle L_1(\Delta f/\epsilon), \cdots, L_d(\Delta f/\epsilon)\rangle, \quad (3)$$

satisfies ϵ-differential privacy, where $Li(\Delta f/\epsilon)$ are i.i.d Laplace variables with scale parameter $\Delta f/\epsilon$.

2.4 EM (Exponential Mechanism)

The Laplacian mechanism only works with numerical data, but sensitive information may also be contained in categorical values, such as the running example in Figs. 1 and 2. Let O be the output of a non-numerical dataset D on function f, and using a scoring function $u(D, r)$ $(r \subseteq O)$, a score is assigned to every output value r. Then, the Exponential Mechanism works as follows

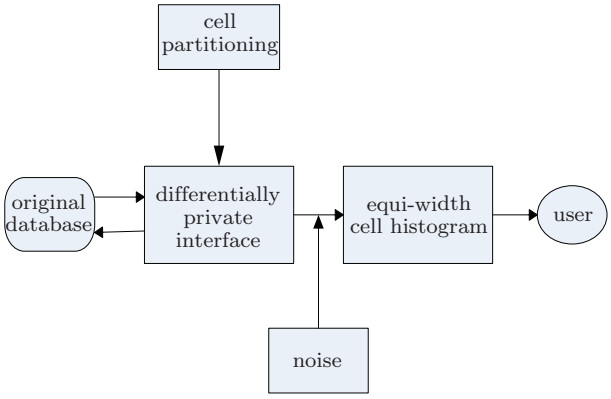

Figure 2 Baseline cell partitioning

Theorem 2 Given a score function $u : (D \times O) \rightarrow R^d$, the mechanism A, which is defined as

$$A(D, u) = \left\{ r : |Pr[r \in O] \propto \exp\left(\frac{\epsilon u(D, r)}{2\Delta u}\right)\right\}, \quad (4)$$

satisfies ϵ-differential privacy, where Δu is the global sensitivity for function u. According to the definition of u, a higher value of u indicates a higher probability for the output r to be selected. This mechanism allows attributes that are non-numerical to be sanitized through the score function u, resulting in a ϵ-DP compliant scheme.

2.5 Differentially private histogram

A histogram is typically defined over a specific domain and a dataset. It summarizes the occurrence counts of domain values in the dataset. For example, if the domain is a set of diseases D (such as cancer, flu, HIV, hepatitis, etc.) then a histogram of a patient dataset would show the number of patients with disease $d \in D$ in the dataset to d. Given the disease information of patients in Fig. 1, Fig. 2 displays a corresponding histogram of the disease attribute. This histogram provides useful statistical summaries of the disease distributions in a given population. However, a histogram inevitably leaks sensitive information from the underlying dataset. For example, if an adversary knows the diseases of all but one patient, they can easily infer the disease of the last patient from the released histogram.

2.6 Two strategies for publishing histogram

Data publication methods based on the differential privacy can be divided into two main strategies:

I. Adding noise to the raw data from statistical information, and optimizing the result. Specifically, adding Laplacian noise to numerical data and exponential noise to nonnumerical data.

II. Start by converting the raw data, and then add noise to the transformed data afterwards.

Methods based on strategy I have better privacy budget with lower utility, and so they do not support high range queries. Strategy II converts the raw data, before adding noise to the results.This can improve data utility and query accuracy, but information may be lost during data transformation.

3 Different methods for publishing differentially private histograms

3.1 Constrained inference

Michael Hay[6] first proposed a method for boosting the accuracy of differentially private histograms

through consistency. The method is called constrained inference. As an example, suppose there is a group of students x_t and the number of students receiving grades x_A, x_B, x_C, x_D, x_F is A, B, C, D and F receptively. The number of students passing is x_p. Suppose all queries need to be answered in away that preserves privacy. One change in the dataset could affect three returned values, therefore, the sensitivity of this set of queries is 3. Additionally, each component will have more noise added by the privacy mechanism. After adding noise, the estimates for x_A, x_B, x_C, x_D, x_F may become worse, but the estimates for x_p may be more accurate. The idea of constrained inference was proposed as a method to solve this inconsistency.

This method uses the input database and privacy parameters to choose a set of queries Q for the data owner. Queries are chosen so that constraints hold among the answers. The data owner then answers the set of queries and adds random independent noise to each answer in the set, and the set of noisy answers is sent to the analyst.

Because the second step is the same as in Ref. [7], this method is able to offer the same differential privacy guarantee.

3.1.1 Histogram segmentation and publication methods based on strategy I

As previously mentioned, experts have developed two methods of segmentation based on strategy I (add noise to original data, and then optimized), There are a baseline cell-based partitioning strategy for releasing an equial-width cell histogram[8] and an innovative 2-phase kd-tree based partitioning strategy for releasing a v-optimal histogram[8].

The first strategy is simple. Based on the original partitioning of the data, a noisy count is released for each cell, resulting in an equal-width cell histogram.

The second strategy is based on the cell-histogram from the first phase and hence exploits the indirectly observed underlying data distribution in the noisy cell histogram. Additionally, the 2-phase method incorporates a uniform measure in the partitioning process, minimizing approximation errors within the partitions. The specific process of the strategy can be described using Fig. 3.

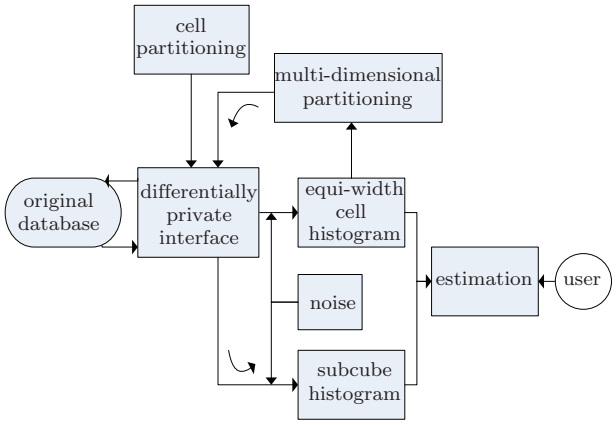

Figure 3 Two-phase partitioning

First, a cell-based partitioning based on the domain is used to generate a fine-grained equal-width cell histogram, which gives an approximation of the original data distribution. Second, a multi-dimensional kd-tree based partitioning is performed to obtain uniform or nearly uniform partitions. Finally, given a user-issued query, an estimation component uses either the v-optimal histogram or both histograms to compute an answer. The advantage of the 2-phase partitioning strategy is that it is both data-aware and adaptive.

3.1.2 Histogram publication methods based on strategy I

Histogram publication methods based on strategy I includes LP[8], Boost1[9] and NoiseFirst[10]. The first method LP was proposed to support unit-length range queries, Laplacian noise is added to each bin directly, this leads to a considerable release error and so it does not support long range queries.

In order to reduce the histogram release error, researchers proposed the other two methods Boost1 and NoiseFirst. Boost1 post-processes an unattributed histogram using the least squares method.

NoiseFirst consists of two steps. In the first step, it calculates a differentially private histogram with the finest granularity, using the Laplacian mechanism with unit-length bins. The sensitivity of the first step is 1, because adding or removing any record can

change the count of any bin by at most 1. Therefore, it is acceptable to inject Laplacian noise into each bin to satisfy ϵ-DP compliance. After the first step, researchers then calculates the optimal histogram structure based on the noisy count sequence $\widetilde{D}$, using dynamic programming algorithm[11]. Apparently, NoiseFirst can be used as a post-processing step to optimize a published histogram $\widetilde{D}$ computed using a Laplacian Mechanism, by merging adjacent noisy counts.

The methods introduced above belong to strategy I. It can be seen from the direction of the research that reduction of errors and support for different types of range queries are the major problems for this type of method.

3.2 Histogram publication methods based on strategy II

Methods from strategy II generally perform better than strategy I as shown in Tab. 2. Thus, researchers proposed a series of methods for publishing differentially private histograms based on strategy II, they can be categorized into four groups.

1) Based on a hierarchical tree structure, the representatives are Privelet[12] and Boost2[9];

2) Based on clustering technology for redrawing each bin of H, the representatives are StructureFirst[10] and P-HPartition[13];

3) Based on Fourier transform compression of H, the representatives are FPA[14] and EFPA[13];

4) Using sensitivity control to publish high-dimensional data, the representatives are DPSense[15] and DPSense-S[15].

3.2.1 Privelet and boost2

Privelet[12] and Boost2[9] have been proposed in order to accurately answer long-range queries. The basis of Privelet is to implement a Haar transform on the original histogram H, converting each bin count into wavelet coefficients and storing them in the middle nodes of the Haar tree. Laplacian noise is added afterwards. The problem with this method is that when the number of bins is high, and the Haar tree is in the index level of the wavelet coefficients, it directly affects the precision and efficiency of a range query.

Boost2 takes advantage of an m-ary tree to reorganize a universal histogram and determine the amount of noise using the height of the tree. After which, Laplacian noise is added to every node in the tree. The method follows the semantic constraints of a range query and the least squares method is used to improve the accuracy of the query response. The downside is that when the number of bins is high, efficiency is low, and this method only supports a one-dimensional histogram.

3.2.2 StructureFirst and P-HPartition

StructureFirst[10] and P-HPartition[13] have been proposed to reduce query sensitivity and improve histogram accuracy. Unlike NoiseFirst, the StructureFirst algorithm computes an optimal histogram structure from the original count sequence, before adding the Laplacian noise to each count. Note that the optimal histogram structure itself is sensitive, therefore, StructureFirst spends a portion of the privacy budget ϵ to protect it. The main idea is to use the v-optimization method to combine near similar bins. The key is choosing the bucket boundaries to avoid information leakage during the combination operation which would be an invasion of privacy. Ref. [16] uses an index mechanism to solve these two problems. Assume that after merging a histogram with m bins, the query sensitivity was $1/p$, noise demand would be $1/P\epsilon_2$, where ϵ_2 denotes the assigned privacy budget for the noise injection phase. Error in this method comes from reconstruction error and perturbation error. Perturbation error consists of indexing noise and Laplacian noise. However, the method does not take the balance between reconstruction error and perturbation error into account, and instead adopts the SSE metric as the main criteria. Error sensitivity is determined by all the upper bound of all bin counts if there are many bins close to the upper bound of the bin count, the SSE method accuracy error is low.

If the histogram contains many bins, adopting StructureFirst will inevitably result in high com-

plexity. P-HPartition uses a greedy bisection strategy to deal with H through top-down segmentation. This continues until each leaf node satisfies predefined stop conditions, and forms m clusters $C_m = \{{\mathbb{C}_1}^m, {\mathbb{C}_2}^m, \cdots, {\mathbb{C}_m}^m\}$. Finally, Laplacian noise is added to the clusters. This method also works on 1-dimensional data, but provides smaller sensitivity.

3.2.3 FPA and EFPA

These two methods compress the data and sanitize the compressed data in order to boost histogram accuracy. The FPA (Fourier Perturbation Algorithm)[15] utilize discrete Fourier transform technology and the Laplacian mechanism to distribute the histogram. Given a histogram $H = \{\mathbb{H}_1, \mathbb{H}_2, \cdots, \mathbb{H}_k\}$, the method applies DFT over H to transform it into Fourier coefficient. As DFT is linearly reversible, each bin can be calculated using the inverse of DFT. The process to apply DFT to histogram H, then choose k coefficients from the resulting F and add Laplacian noises to them, the final step is to append $n-k$ zeros to the histogram and perform the inverse Fourier transformation. After these operations have been performed, the histogram can be released. The error in FPA consists of reconstruction error and noise error. Reconstruction error is caused by neglected $n-k$ coefficients, which are substituted with zeros. The choice of k is very important. A larger k increases perturbation error, and a smaller k increases reconstruction error. In order to choose the most appropriate k, Rastogi and Nath[17] also proposed an extension to FPA, called the Sampling Perturbation Algorithm (SPA), to select k adaptively depending on the input dataset. SPA uses exponential mechanism to select a value of k by the utility function $U(H, K)$, where

$$U(H,K) = \left(\sum_{i=k+1}^{n} \mid F_{i-1} \mid^2 \right)^{\frac{1}{2}} + \frac{k\sqrt{n}}{\epsilon}. \quad (5)$$

EFPA[13] is an enhanced version of FPA defined in Ref. [7]. Suppose histogram H has length n, where n is odd. As EFPA applies DFT on H, it operates with m instead of n coefficients. Finally, EFPA perturbs each ${F_i}^k$. Note that the number of retained coefficients is $z = 2k+1$. This is because other than ${F_0}^k$ the complex conjugate of all other coefficients also appears in the complete Fourier transform of H. Finally, $\widetilde{F^k} = F^k + (L(2\sqrt{z}/\epsilon))^k$ is assembled as an m-dimensional vector by appending $m-k$ zeros. EFPA substantially reduces the amount of noise added.

3.2.4 DPSense and DPSense-S

For high-dimensional datasets, the key challenge is their high sensitivity. Large amounts of noise must be added to satisfy differential privacy. In these scenarios, each tuple of the input dataset D is a binary vector in $\{0,1\}^d$, and d can be very large. DPSense is an approach for publishing statistical information from datasets with differential privacy via sensitivity control. DPSense-S is a scalable version of DPSense.

The key idea of DPSense is to choose an appropriate sensitivity threshold θ and to limit each row's contribution to the column count vector to at most θ. Given a threshold θ, the method for limiting sensitivity is normalization: $D|_\theta(i,j) = D(i,j)$, where $RC_i < \theta$; $\theta(i,j) = D(i,j)\dfrac{\theta}{RC_i}$, where $RC_i > \theta$.

In order to minimize the effect of truncation errors caused by the normalization step and noise errors caused by the addition of Laplacian noise, the most important problem is choosing the best θ. This is because increasing θ reduces truncation errors while increasing noise errors. According to the exponential mechanism, the optimal quality function is

$$q(D,\theta,\epsilon_p) = ac(D|\theta) - \frac{\theta}{\epsilon_p}, \quad (6)$$

where $Prob[\theta]$ satisfies ϵ_c-differential privacy. The design principle of the quality function is that noise error can be represented by $\dfrac{\theta}{\epsilon_p}$, and the truncation error can be represented by $ac(D|\theta)$.

Because column counts from a normalized dataset always underestimate the true counts, DPSense has a systematic under-estimation bias. In order to address the problem, an effective method is to correct this underestimation by scaling up the noisy counts

output by DPSense with a factor α, then the quality function is changed to

$$qs(D, \theta, \alpha, \epsilon_p) = -ae(D, D|\theta) - \alpha\frac{\theta}{\epsilon_p}. \quad (7)$$

Experimental evaluation using several utility metrics demonstrates the effectiveness of publishing all columns, and the efficiency of these algorithms makes private publishing of high dimensional datasets more practical.

4 Conclusion

Now that the various differentially private histogram publication methods have been outlined above, we summarized and compared the methods by their different properties in Tab. 2 below.

The development of differentially private histogram publication methods has drawn many research efforts and has produced many powerful approaches. Because different application requirements produce different data characteristics, there are still many problems that require further in-depth research.

5 Expansion

In the methods discussed above, the released histogram is only applicable to static data. In the real world, there is a large amount of real-time data that must be studied and summarized. Unlike static data which can be studied easily, dynamic data is constantly changing. A series of approaches have been proposed to capture dynamic data. We studied the most representative methods DSFT and DSAT. When we receive sample data from a dynamic database in real time, it can be used as a black box for generating "one time" histograms. Next, we will outline the two algorithms.

DSFT (Distance-based Sampling with Fixed Threshold) works to determine a value T. assuming that there is an optimal value of T which will enable the algorithm to exactly generate exact differentially private histograms. If the threshold T is higher than the optimal value, there will be a leftover privacy budget that is not funny utilized. Conversely, if T is smaller than the optimal value, the privacy budget will be exhausted prematurely, resulting in update errors for the remaining time points. DSFT uses a fixed threshold and is divided into two steps at each time point t_i: decision and sampling. The decision step computes a noisy distance between the original dataset H_i at the current time stamp and the latest released histogram $\widetilde{H_j}$ and determines if it is larger than the noise threshold $\widetilde{T}$. If it is, then the sampling step generates a new differentially private histogram $\widetilde{H_i}$, otherwise it outputs the previous $\widetilde{H_j}$. The overall privacy budget is divided between the decision and sampling steps which are designed to guarantee differential privacy as we will discuss later.

DSAT (Distance-based Sampling with Adaptive Threshold) releases a series of differentially private dynamic histograms while adaptively adjusting the threshold T_i at each time point, based on evolution in the raw data. With DSAT, researchers do not need to find an optimal value of T which may be difficult in practice. We use T_i to denote the generated threshold at t_i and the other notations are the same as DSFT. T_1 is set to be $T + Lap(\Delta/\widetilde{\epsilon_1})$, $\widetilde{\epsilon_1}$ is a tiny privacy budget because the initial value T_1 is not significant in DSAT. We only need to constrain it between 0 and 2, which is the domain of the L_1 distance. Next, $\widetilde{D_1}$ is used for the first M time points, where M is a small integer number to allow a burn-in period for discrepancy to be accumulation, avoiding frequent updates of T_i during early time periods. M can be user-specified and is not a sensitive parameter, it only needs to be much smaller than N (Number of time points).

By comparing the two methods, we draw the conclusion that the error of DSFT is very sensitive to the threshold value T[1]. As T increases initially, the error decreases because of the decreased perturbation error. As T increases further, the error increases due to the increased sampling error which becomes the dominant error. Without prior knowledge, the optimal T is difficult to determine. However, the average absolute error of DSAT is close to the lowest error of DSFT with the optimal threshold value T being approximately 0.025[1]. The initial value of T for

Table 2 Histogram release strategy comparation

method		Ref. [7]	Ref. [9]	Ref. [10]	Ref. [12]	Ref. [13]	Ref. [10]	P-HP	Ref. [15]	EFPA	DS	DS-S
conversion technology	tree transform					✓						
	wavelet transform				✓							
	Fourier transform								✓	✓		
	cluster transform			✓			✓	✓				
dimension	1	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	$\geqslant 2$				✓				✓		✓	✓
query	long range count		✓	✓	✓	✓	✓	✓			✓	✓
	unit count	✓	✓		✓	✓			✓	✓		
data relationship	independent	✓	✓		✓	✓			✓	✓	✓	✓
	dependent			✓			✓	✓				
error boundary	linear level	✓					✓	✓	✓	✓	✓	✓
	log level		✓	✓	✓	✓						
error source	reconstruction error			✓			✓	✓	✓	✓		
	perturbation error	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
data type	static	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	dynamic											

DSAT can be selected arbitrarily. Thus, the DSAT method can effectively adjust T to its optimal value.

References

[1] C. Li, G. Miklau, M. Hay, et al. The matrix mechanism: optimizing linear counting queries under differential privacy [J]. The VLDB journal, 2015, 24(6): 757-781.

[2] C. Y. Song, T. J. Ge. Aroma: a new data protection method with differential privacy and accurate query answering [C]//The 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 2014: 1569-1578.

[3] G. Z. Yuan, Z. J. Zhang, M. Winslett, et al. Optimizing batch linear queries under exact and approximate differential privacy [J]. ACM transactions on database systems, 2015, 40(2): 11.

[4] J. Xu, Z. J. Zhang, X. K. Xiao, et al. Differentially private histogram publication [J]. The VLDB journal, 2012, 22(6): 32-43.

[5] H. R. Li, X. Li, X. Q. Jiang, et al. Differentially private histogram publication for dynamic datasets: an adaptive sampling approach [C]//The 24th ACM International Conference on Information and Knowledge Management, Melbourne, Australia, 2015: 1001-1010.

[6] M. Hay, M. Miklau, G. Miklau. Boosting the accuracy of differentially private histograms through consistency [J]. Proceedings of the VLDB endowment, 2010, 3(1-2): 1021-1032.

[7] C. Dwork, M. Mcsherry, K. Nissim, et al. Calibrating noise to sensitivity in private data analysis [C]//The 3rd Theory of Cryptography Conference, New York, USA, 2006: 265-284.

[8] Y. H. Xiao, X. Li, Y. Y. Fan, et al. DPCube: differentially private histogram release through multidimensional partitioning [J]. Transactions on data privacy, 2014, 7(3): 195-222.

[9] M. Hay, V. Rastogi, G. Miklau, et al. Boosting the accuracy of differentially private histograms through consistency [C]//The 36th Conference of Very Large Databases (VLDB), Istanbul, Turkey, 2010: 1021-1032.

[10] J. Xu, Z. J. Zhang, X. K. Xiao, et al. Differentially private histogram publication [C]//IEEE 28th International Conference on Data Engineering (ICDE), Washington, DC, USA, 2012: 32-43.

[11] H. V. Jagadish, N. Koudas, S. Muthukrishnan, et al. Optimal histograms with quality guarantees [C]//The 24th International Conference on Very Large Data Bases, New York, USA, 1998: 275-286.

[12] X. K. Xiao, G. Z. Wang, J. Gehrke. Differential privacy via wavelet transforms [J]. IEEE trans knowl data eng, 2011, 23(8): 1200-1214.

[13] G. Acs, C. Castelluccia, R. Chen. Differentially private histogram publishing through lossy compression [C]//The 12th IEEE International Conference on Data Mining (ICDM), Brussels, Belgium, 2012: 84-95.

[14] V. Rastogi, S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption [C]//The ACM SIGMOD International Conference on Management of Data (SIGMOD), Indianapolis, USA, 2010: 735-746.

[15] W. Y. Day, N. H. Li. Differentially private publishing of high-dimensional data using sensitivity control [EB/OL]. http://dx.doi.org/10.1145/2714576.2714621.

[16] C. Dwork. Differential privacy [C]//The 33rd International Colloquium on Automata, Languages and Programming (ICALP), Venice, Italy, 2006: 1-12.

[17] V Rastogi, S Nath. Differentially private aggregation of distributed time-series with transformation and encryption [C]//2010 ACM SIGMOD/PODS Conference, Indianapolis, USA, 2010: 735-746.

About the authors

Xue Meng received B.Eng. from Xi'an University of Post and Telecommunications in 2014. She is currently a graduate student in the School of Cyber Engineering at Xidian University. Her research insterests include privacy preserving data management and differentially private data publication. (Email: 1067901461@qq.com)

Hui Li [corresponding author] received the B.Eng. from the Harbin Institute of Technology in 2005 and Ph.D. degree from Nanyang Technological University, Singapore in 2012, respectively. He is an associate professor in the School of Cyber Engineering, Xidian University, China. His research interests include data mining, knowledge management and discovery and privacy-preserving queries and analysis in big data. He has over 30 publications in data management research, the majority of which appear in top-tier venues such as SIGMOD, SIGKDD, VLDB, ICDE, INFOCOM, TKDE and VLDB Journal. (Email: hli@xidian.edu.cn)

Jiangtao Cui received the M.S. and Ph.D. degrees both in computer science, from Xidian University, Xi'an, China in 2001 and 2005 respectively. Between 2007 and 2008, he was with the Data and Knowledge Engineering group working on high-dimensional indexing for large scale image retrieval, at the University of Queensland (Australia). He is currently a professor in the School of Cyber Engineering, Xidian University, China. His current research interests include data and knowledge engineering, and high-dimensional indexing. (Email: cuijt@xidian.edu.cn)