A One-Pass Distributed and Private Sketch for Kernel Sums with Applications to Machine Learning at Scale

0. Abstract

大多数的 private algorithm 强调解决的是 task-dependent functions。

提出 private sketch，适用于 large-scale

核心: the reduction to kernel sums.

sketch 估计这些 sums，使用 locality-sensitive hashing，随机tables

现存的 kernel sum estimation 是对于 scale poorly，对于dimension 的增加，会变得非常慢

但是，本文提出的 sketch 可以运行在 large high-dimensional datasets.

在 privacy-utility tradeoff 与现有的方法差不多，但是计算开销得到了数量级的减小。

1. Introduction

Differential privacy 可以提供一个 minimum level of protection.

保护的强度是通过 ϵ 来测量的

small value ϵ 表示泄露很少的信息

Private Function Release

- a pairwise operation k(x,q)
- query q
- dataset $D = x_1, x_2, \ldots x_N$
- objective:

计算 ϵ-differential private version

$$f_D(q) = sum_{x \in D} k(x, q)$$

即使直接 evaluate $f_D(q)$，对于 each query 也会 leak additional information.

在 privacy budget runs out之前，只能有限次数 evaluate $f_D(q)$

因为这个 limitation，考虑 private function release problem.

并不是通过增加 privacy budget 来增加 queries，而是通过 privacy budget 来发布 a private summary S_D of f_D

summarization algorithm 是 private --> queries made by summary 不会额外消耗 privacy budget。

Application to Private Machine Learning

Practical Limitations

现有的方法只是适用于 small scale, low-dimensional datasets.

实际的 function release 受限于 runtime and memory requirements.

即使对于 low-dimensional，这个过程也需要几天时间

其他的方法是需要 large kernel matrices 特征值分解 或者对于每个 query 的Monte Carlo 插值。

很多方法都不适用于 dimensions > 3的情况

现有的方法不满足 practical 对于 large-scale demands

Scalable Function Release with Sketches

steaming algorithms and compressed sketches.

goal: 解决互联网规模的问题中的一次性数据传输

Function release method 是 RACE sketch 的扩展

继承了 RACE 的许多属性，eg. large-scale, low communication overhead, ϵ-differential privacy.

实验证明，算法具有很小的 computation cost

Sketches for Kernel Compositions

用 flexibility 换取了 computational efficiency.

sketch 限制了 f_D 的选择到一个特定的 kernel class k(x,q)，即 locality sensitive hash (LSH) kernels.

RACE 可以实现在 large, high-dimensional datasets 上运行的需求

Our Contribution

- 提出 private version of RACE sketch
- 证明了 error bounds
- 将几类重要的ML技术简化为 LSH 的近似和
- 对于3种应用实验

  density estimation

  regression

  classification
- 对于 large-scale kernel sum tasks 得到了数量级的加速

  kernel density tasks 之前需要数小时的任务，使用 private RACE sketch只需要几秒钟即可

2. Background

2.1 Differential Privacy

Definition 2.1. Differential Privacy [15] A randomized function A is said to provide (ϵ, β)-differential privacy if for all neighboring databases $\mathcal{D}$ and $\mathcal{D}'$ (which differ in at most one element) and all subsets S in the codomain of A,

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

ϵ 是 privacy budget，限制了 function可以泄露的 individual information

β 如果大于0，那么 function 潜在可能泄露的信息的概率会增加 β

本文，设置了 $\beta = 0$ ，即 ϵ-differential privacy

Laplace 机制满足上述条件

通过给 real-valued function 添加了 zero-mean Laplace noise，如果 noise 基于function 的 sensitivity，则满足 DP

Definition 2.2. Sensitivity [15] For a function $A : \mathcal{D} \to \mathbb{R}^d$, the L1-sensitivity of A is

$$\Delta = \sup ||A(\mathcal{D}) - A(\mathcal{D}')||_1$$

where the supremum is taken over all neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$

THEOREM 2.3. *Laplace Mechanism [15] Let $A : \mathcal{D} \to \mathbb{R}^d$ be a non-private function with sensitivity Δ and let $\mathbf{z} \sim \text{Lap}(\Delta/\epsilon)^d$ (d-dimensional i.i.d Laplace vector). Then the function $A(\mathcal{D}) + \mathbf{z}$ provides ϵ-differential privacy.*

2.2 Related Work

Release the kernel sum f_D 的几种方法的 error and computational complexity.

Method	Error Bound	Runtime	Comments
Bernstein polynomials [3]	$d\frac{d}{d+H}\left(\frac{1}{\epsilon}\log\frac{1}{\delta}\right)^{\frac{H}{d+H}}$	$O(dNM^d)$	$M \geq 2$. Memory is also exponential in d.
PFDA [31]	$\frac{2}{\epsilon}\sqrt{\log\frac{2}{\beta}\log\frac{1}{\delta}\frac{C}{\phi}}$	$O(dN^2)$	C and ϕ are task-dependent (ϵ, β)-differential privacy
MWEM [23]	$N^{\frac{2}{3}}\left(\frac{\log N \log\lvert Q\rvert}{\epsilon}\right)^{1/3}$	$O(dN\lvert Q\rvert)$	Q is a set of query points. Holds with probability $1 - 1/\text{poly}(\lvert Q\rvert)$
Trigonometric polynomials [45]	$\frac{1}{\epsilon}N^{\frac{2d}{2d+H}}$	$O(dN^{1+\frac{d}{2d+H}})$	The result holds with probability $1 - \delta$ for $\delta \geq 10e^{-\frac{1}{5}N^{d/2d+K}}$
This work	$\left(\frac{N}{\epsilon}\log\frac{1}{\delta}\right)^{\frac{1}{2}}$	$O(dN)$	Applies only for LSH kernels. Efficient streaming algorithm.

Table 1: Summary of related methods to release the kernel sum $f_{\mathcal{D}} = \sum_{\mathcal{D}} k(\mathrm{x}, \mathrm{q})$ for an N point dataset $\mathcal{D}$ in $\mathbb{R}^d$. Unless otherwise stated, the error is attained with probability $1 - \delta$ and ϵ-differential privacy. We hide constant factors and adjust results to estimate $f_{\mathcal{D}}$ rather than the KDE ($N^{-1}f_{\mathcal{D}}(\mathrm{q})$) when necessary. H is a kernel smoothness parameter.

2.3 Locality-Sensitive Hashing

LSH Functions

LSH family of functions $l(x)$ F，具有如下属性：

（1）在 $l(x)$ 下，相似的 point 有很高的概率会有相同的 hash value

当 两个 points 有相同的 hash code， $l(x) = l(y)$，则称发生了collision

Definition

hash family 是 locality-sensitive，有 collision 概率是 $k(\cdot,\cdot)$，在从 LSH family中 uniform random 选取的情况下，$l(x) = l(y)$ 的概率是 k(x,y)

LSH Kernels

collision 概率 k(x,y) 满足符合 dist(x,y) 单调递减函数

k 是 positive semidefinite kernel function.

kernel function k(x,y) 是 LSH kernel

2.4 RACE Sketch

$$f_D(q) = \sum_{x \in D} k(x, q)$$

k(x,q) 是一个 LSH kernel

已有的work 提出了一个 one-pass streaming algorithm去估计 kernel density sums.

a RACE (Repeated Array of Count Estimators) sketch $S_D \in Z^{R \times W}$

从 LSH family F，以 desired collision 概率构造 R个 functions $\{l_1(x), l_2(x) \ldots l_R(x)\}$

对于 element x，hash x 得到 R 个 hash values，即 S_D的每一行。对于 dataset 的每个 element 重复上述过程。

在近似$f_D(q)$ 的时候，返回 R 行的$S_D[r, l_r(q)]$的均值

在证明 error bounds 的同时，S_D 的每一行是 f_D 的无偏估计

THEOREM 2.5. *Unbiased RACE Estimator[12] Suppose that X is the query result for one of the rows of $\mathcal{S}_\mathcal{D}$. That is, $X = \mathcal{S}_\mathcal{D}[r, l_r(\mathbf{q})]$*

$$\mathbb{E}[X] = f_\mathcal{D}(\mathbf{q}) \qquad \text{var}(X) \leq \left(\tilde{f}_\mathcal{D}(\mathbf{q})\right)^2 = \left(\sum_{\mathbf{x} \in \mathcal{D}} \sqrt{k(\mathbf{x}, \mathbf{q})}\right)^2$$

3. Private Sketches With RACE

提出了 RACE sketch 的private version

给 RACE sketch array 的每个 count 应用了 Laplace 机制

Algorithm 1 介绍了 differentially private method to release RACE sketch

Algorithm 1需要 $O(NR)$ hash computations.

Algorithm 1 Private RACE sketch

Input: Dataset $\mathcal{D}$, privacy budget ϵ, LSH family $\mathcal{F}$, dimensions $R \times W$

Output: Private sketch $\mathcal{S}_\mathcal{D} \in \mathbb{Z}^{R \times W}$

Initialize: R independent LSH functions $\{l_1, \ldots, l_R\}$ from the LSH family $\mathcal{F}$

$\mathcal{S}_\mathcal{D} \leftarrow \mathbf{0}^{R \times W}$

for $\mathbf{x} \in \mathcal{D}$ **do**

  for r in 1 to R **do**

    Increment $\mathcal{S}_\mathcal{D}[r, l_r(\mathbf{x})]$

  end for

end for

$\mathcal{S}_\mathcal{D} = \lfloor \mathcal{S}_\mathcal{D} + Z \rfloor$ where $Z \overset{\text{iid}}{\sim} \text{Lap}(R\epsilon^{-1})$

RACE sketch 的流程如下：

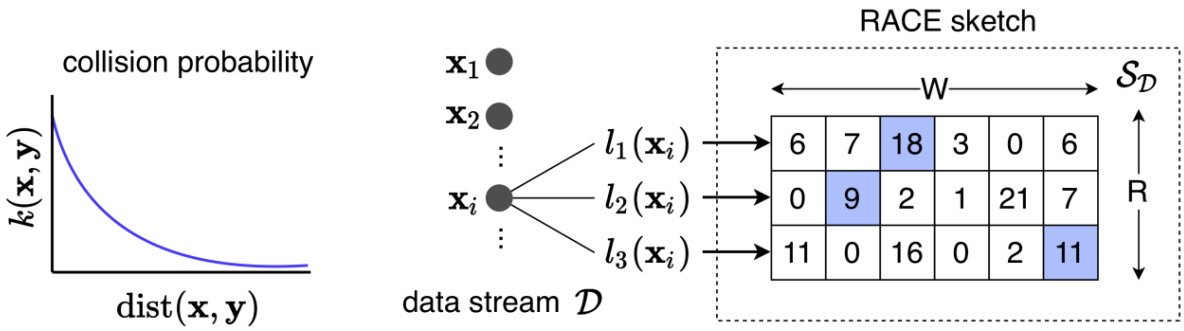

Figure 1: Illustration of Algorithm 1 for $\mathcal{S}_\mathcal{D} \in \mathbb{Z}^{3\times6}$. We hash each element in the stream with LSH functions $\{l_1, l_2, l_3\} \in \mathcal{F}$ having collision probability $k(\mathbf{x}, \mathbf{y})$. In this example, $l_1(\mathbf{x}_i) = 3$, $l_2(\mathbf{x}_i) = 2$ and $l_3(\mathbf{x}_i) = 6$. We increment the highlighted cells. The addition of the Laplace noise is not shown in the figure, but is done by perturbing each count in $\mathcal{S}_\mathcal{D}$.

Algorithm 2显示了 query the sketch

column $l_r(q)$ value 是 $f_D(q)$ 的无偏估计

Algorithm 2 是简单地平均了 R 个 estimators.

Algorithm 2 RACE query

Input: Sketch $\mathcal{S}_{\mathcal{D}}$, query $\mathbf{q}$, the same R LSH functions from Algorithm 1
Output: Estimate of $N^{-1} f_{\mathcal{D}}(\mathbf{q})$
$\widehat{N} \leftarrow R^{-1} \sum_{i,j} \mathcal{S}_{\mathcal{D}}[i,j]$ (optional normalization step)
$\hat{f}_{\mathcal{D}} \leftarrow 0$
for r in 1 to R **do**
  $\hat{f}_{\mathcal{D}} = \hat{f}_{\mathcal{D}} + \frac{1}{R}\mathcal{S}_{\mathcal{D}}[r, l_r(\mathbf{q})]$
end for
Return: $\hat{f}_{\mathcal{D}}/\widehat{N}$

3.1 Privacy

证明 Algorithm 1 的 返回结果是 ϵ differentially private

Function A $D- > R^{R\times W}$，输出了 $R \times W$ RACE sketch.

LEMMA 3.1. *Consider one row of the RACE sketch, and add independent Laplace noise* $\mathrm{Lap}(1/\epsilon)$ *to each counter. The row can be released with* ϵ*-differential privacy.*

3.2 Utility

LEMMA 3.3. *Let* $X_1, \ldots X_R$ *be* R *i.i.d. random variables with mean* $\mathbb{E}[X] = \mu$ *and variance* $\leq \sigma^2$*. To get the median of means estimate* $\hat{\mu}$*, break the* R *random variables into* k *groups with* $m = R/k$ *elements in each group.*

$$\hat{\mu} = \text{median}\left(\frac{1}{m}\sum_{i=1}^{m} X_i, \ldots, \frac{1}{m}\sum_{i=(k-1)m+1}^{km} X_i\right)$$

Lemma 3.3 requires a bound on the variance of the private RACE estimator, which we obtain by adding the independent Laplace noise variance $2R^2\epsilon^{-2}$ to the bound from Theorem 2.5. Theorem 3.4 follows.

THEOREM 3.4. *Let $\hat{f}_{\mathcal{D}}(\mathbf{q})$ be the median-of-means estimate using an ϵ-differentially private RACE sketch with R rows and $\tilde{f}_{\mathcal{D}}(\mathbf{q}) = \sum_{\mathcal{D}} \sqrt{k(\mathbf{x}, \mathbf{q})}$. Then with probability $1 - \delta$,*

$$|\hat{f}_{\mathcal{D}}(\mathbf{q}) - f_{\mathcal{D}}(\mathbf{q})| \leq \left(\frac{\tilde{f}^2_{\mathcal{D}}(\mathbf{q})}{R} + \frac{2}{\epsilon^2} R \right)^{1/2} \sqrt{32 \log 1/\delta}$$

3.3 Practical Implications

使用的是 average 或者 median-of-means estimator

Low-Density Queries

low-density regions appear to be noisier.

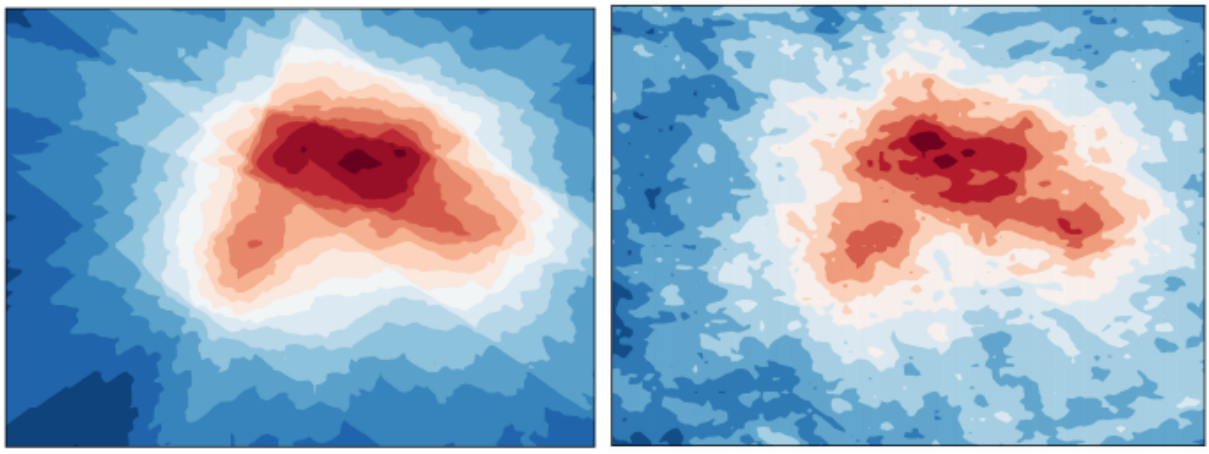

Figure 2: Visualization of non-private (left) and private (right) sketch-based KDE. Note the distinct boundaries between regions that map to different locations and counts in the array. In $\mathbb{R}^d$, RACE approximates functions with a piecewise-constant spline over random partitions. The Laplace mechanism perturbs the value of each partition.

Hyperparameter Tuning

必须花费一定份额的 privacy budget，去评估每个超参数的组合。

sketch 有 3 个超参数，W，R，kernel k(x,q)

kernel 是事先知道的，W 是能从 Kernel 中确定的，Corollary 3.5 表明，optimal value $R = \lceil \frac{1}{\sqrt{2}} \widetilde{f}_D(q)\epsilon \rceil$

但是，无法知道 $E_q[\widetilde{f}_D(q)]$，可以从 $[1, \frac{1}{\sqrt{2}}N\epsilon]$ 选取一个合适的 R 值

4. Applications

应用于 density estimation, classification, regression, mode finding, anomaly detection and sampling.

Kernel Density Estimation

kernel density estimation 是一个经典的非参数的方法，可以直接估计 dataset 的分布

Mode Finding

给定了概率密度，可以 locate data distribution

gradient-free 在sketch的优化很好，但是 KDE 是一个 non-convex function

LSH kernel 可以用来估计比正半定kernel 更多样化的函数集

5. EXPERIMENTS

比较 KDE、classification、regression、

KDE 估计了 salaries density 对于 New York City (NYC) and San Francisco(SF) city employees in 2018。以及 high-dimensional densities for the skin, codrna and covtype UCI datasets.

对于 KDE task，由于 data size 的原因，不能直接运行，需要采用 sampling and dimensionality reduction 才可以与 baseline 比较。

对于 regression and classification 实验，使用的是 UCI 数据集

Dataset	N	d	σ	Description	Task
NYC	25k	1	5k	NYC salaries (2018)	KDE
SF	29k	1	5k	SF salaries (2018)	
skin	241k	3	5.0	RGB skin tones	
codrna	57k	8	0.5	RNA genomic data	
covtype	580k	55	20	Cartographic features for forestry	
nomao	34k	26	0.6	User location data	Classification
occupancy	17k	5	0.5	Building occupancy	
pulsar	17k	8	0.1	Pulsar star data	
airfoil	1.4k	9	-	Airfoil parameters and sound level	Regression
naval	11k	16	-	Frigate turbine propulsion	
gas	3.6k	128	-	Gas sensor, different concentrations	

Table 2: Datasets used for KDE and classification experiments. Each dataset has N entries with d features. σ is the kernel bandwidth.

Hyperparameters

对于 kernel density estimation and max likelihood classification，使用 p-stable Euclidean LSH kernel

比较算法在最优参数下的 performance and computation cost。

5.1 Baseline Algorithms

Spectral Approximation

使用 Fourier series近似 kernel density function，truncate 序列展开，给每个系数添加 Laplace noise

光谱近似适用于数据集在 [0,1] 之间

Bernstein Mechanism

基于函数近似

但是使用的是 Bernstein polynomials 而不是 Fourier basis。

Private Functional Data Analysis (PFDA)

保证了 (ϵ, δ) differential privacy, 使用的是 Gaussian mechanism

Kernel Mean Embedding (KME)

通过 KME release 的数据对合成数据进行加权。

合成public 数据库的 kernel mean embedding 接近于 private database

RACE

W = 1000

使用程序选取 R

对于 classification 实验，使用 RACE 构造了 max-likelihood classifiers 去估计似然函数

5.2 Results

如图是KDE experiments，使用L_2 function norm 来近似 KDE 和 truth KDE

对 grid 上的error 进行积分，来近似估计 L_2 function error

图中可以看出 error 和 privacy budget 的关系

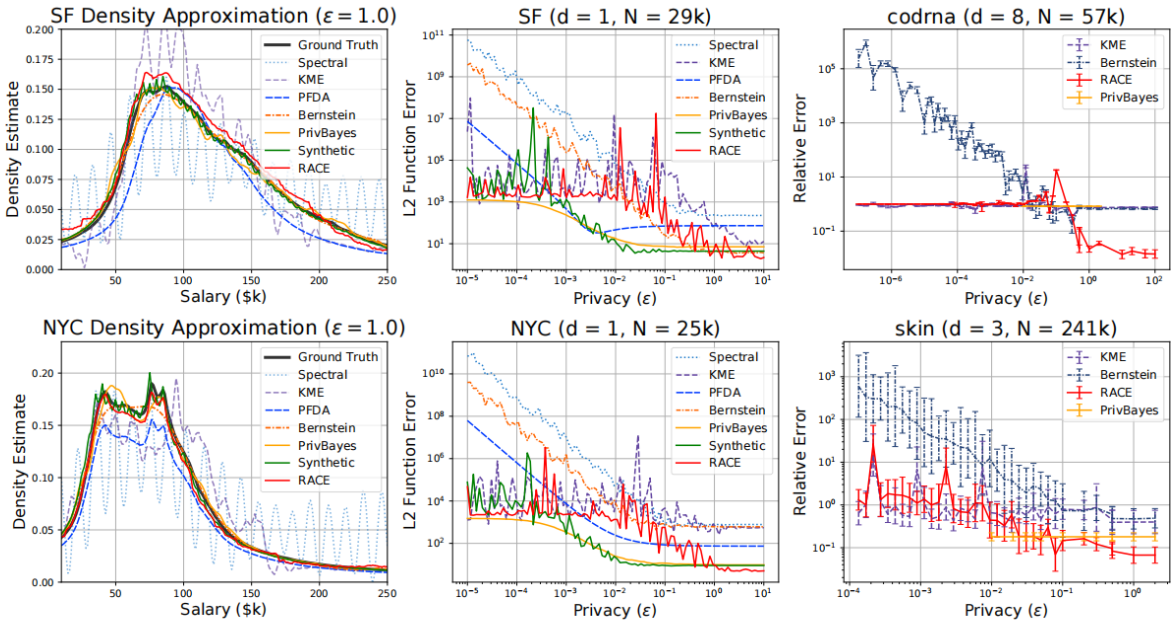

Figure 3: Privacy-utility tradeoff for private function release methods. We report the L2 function error and the mean relative error for 2000 held-out queries.

下图展示的是分类实验的结果：

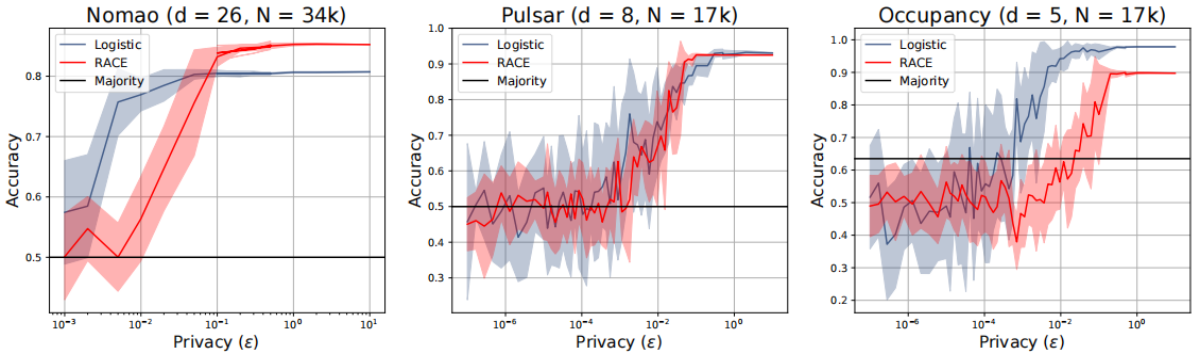

Figure 4: Binary classification experiments. We show the privacy-utility tradeoff for a private logistic regression classifier and the RACE max-likelihood classifier. Average over 10 repetitions.

下图展示的是线性回归实验的结果：

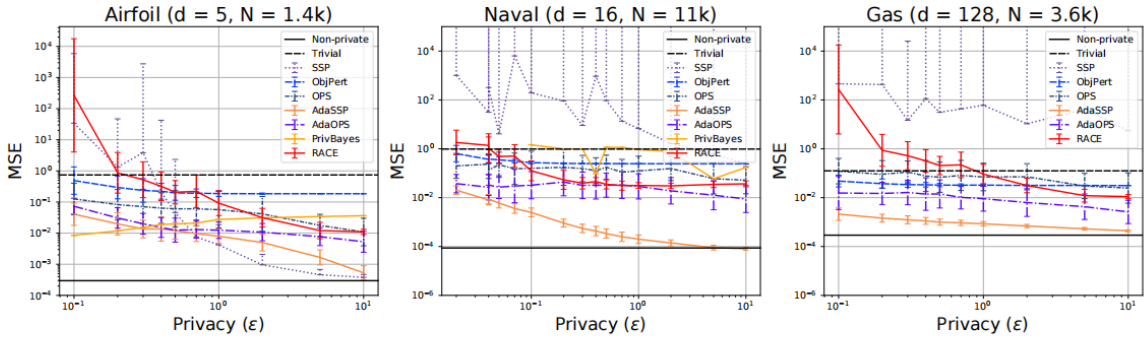

Figure 5: Linear regression experiments. We show the privacy-utility tradeoff for RACE and several other linear regression methods. Average over 10 repetitions.

下表展示的是构造 function release 的计算时间

	PFDA	Bernstein	KME	RACE	PrivBayes
Sketch	> 3 days	2.3 days	6 hr	15 sec	12 sec
Query	-	6.2 ms	1.2 ms	0.4 ms	0.5 sec

Table 3: Computation time for KDE on the skin dataset. PFDA was unable to finish on this dataset within 3 days.

6. Large Scale Experiments

Friendster graph 需要数天的时间，但是本文的 sketch 算法只需要 18 minutes

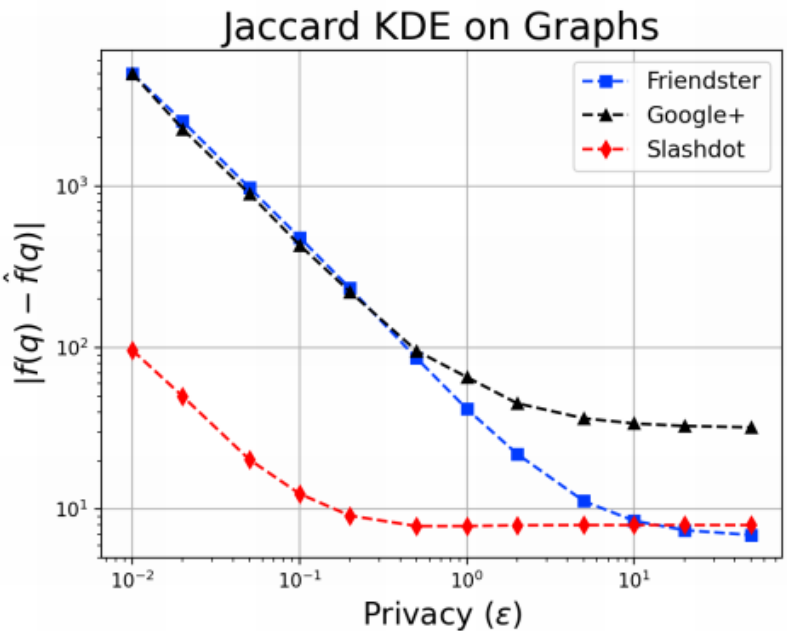

Figure 6: Approximating the Jaccard KDE over large social network graphs. We use $R = 2\text{k}$ and $W = 100\text{k}$ except for Friendster, where we use $W = 500\text{k}$.

graph 实验 的运行时间

Graph	N	Edges	Sketch	Query	Baselines
Friendster	65M	1.8B	18.1 min	164 μs	> 1 week
Google Plus	72.3K	13M	31 sec	238 μs	> 1 week
Slashdot	82.2K	948K	25 sec	157 μs	> 1 week

Table 4: Large-scale datasets used for graph experiments. The sketch time is reported for our parallel sketch implementation. Query times are an average over ≈2k queries. We were unable to run baseline methods on problems of this size, but based on rough calculations we estimate that it would take weeks to perform function release on these datasets with baseline algorithms.

7. Extensions

7.1 Learned Hash Constructions

扩展 RACE 到更多的 values of k(x,q)

(1) construct a new ker nel by combining existing LSH functions

(2) adopt methods from the learning to hash literature

7.2 Access to Public Data

recent works 用来使用 publicly-available information 去提升 private queries.

例如：Public-Assisted Private framework

improve our max-likelihood classifier

learning 一个 hash function，导致相同类型的 class 会 collide 更频繁

7.3 Private Distributed Sketching

3 个重要属性：

（1）sketch 需要的内存主要取决于期望的近似误差，对N是 sublinear

（2）sketch 可以高效构造通过 a single pass

（3）sketches 是可以合并的，对于 dataset D_1 和D_2 ，可以获得一个 sketch 的组合数据 $D_1 \cup D_2$，只需要简单地相加 counters S_1 and S_2

在 private 方面，得益于federated learning，很多的 private aggregation 的方

- multi-party protocols
- local noise addition

Local Noise Addition

adding Gaussian or Binomial noise 来满足 $(\epsilon, \beta)-$differential privacy

可以直接应用于 sketch

8. Discussion

RACE 只估计 LSH kernels，LSH kernels 的空间足够大，可用于 machine learning

private RACE sketch 对于真实应用依旧是可行的

9. Conclusion

- 提出了一个 differentially private sketch，适用于大量的 machine learning tasks
- RACE 适用于 density estimation, classification, linear regression.
- sketches 可在分布式 one-pass streaming 中设置，并且高度计算高效