UNIVERSITÄT ZU LÜBECK

Answering Many Database Queries With High Privacy and Precision

D. Breutigam, R. Reischuk

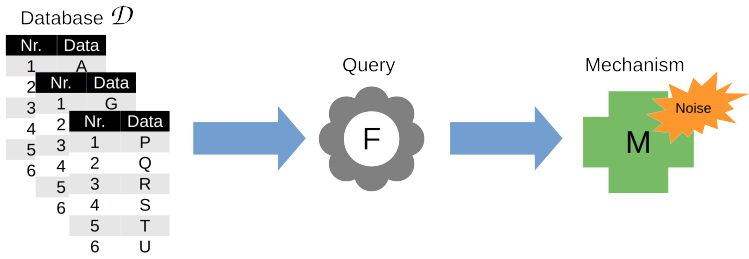
18/12/25

IM FOCUS DAS LEBEN



How to Model Privacy

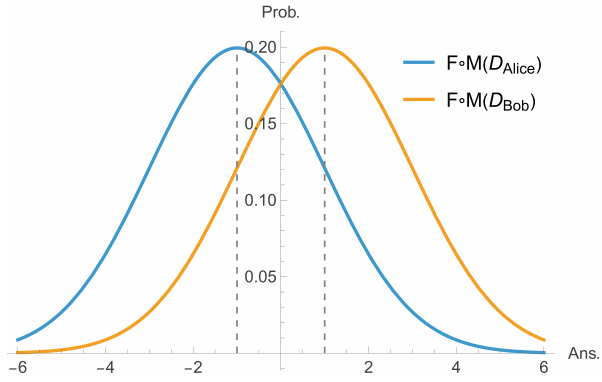
Differential Privacy Model



How to Model Privacy

Privacy thread is modelled as the ability of an adversary to differentiate whether a specific entry of the database D contains is $Alice$ or Bob based on answers of queries.

How to Differentiate



Indistinguishability

For any set S of possible answers

$$\mathbb{P}\left[M \circ F(D_{\text{Alice}}) \in S\right] \leq e^{\varepsilon}\, \mathbb{P}\left[M \circ F(D_{\text{Bob}}) \in S\right] + \delta.$$

Indistinguishability

For any set S of possible answers

$$\mathbb{P}\left[M \circ F(D_{\text{Alice}}) \in S\right] \leq e^{\varepsilon}\, \mathbb{P}\left[M \circ F(D_{\text{Bob}}) \in S\right] + \delta.$$

Common Privacy Parameters

- ϵ bounds the increase of the probability mass
- δ bounds the probability of large deviations

How to Model Limited Background Knowledge

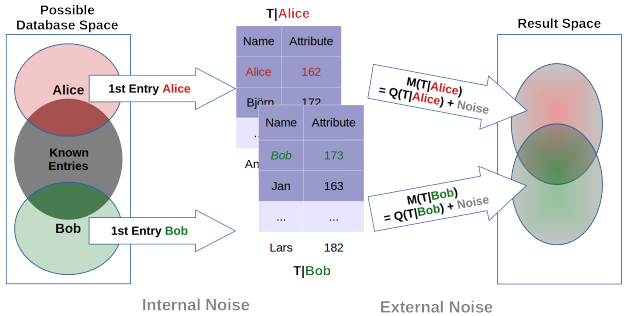
- examine the actual data
- consider public information
- determine the most likely distributions
- choose the worst possible privacy parameters

Why?

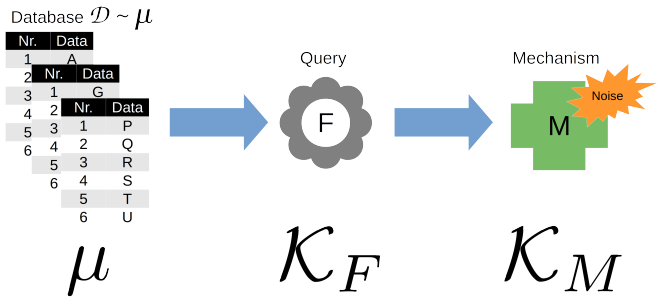
$\Rightarrow$ corresponds to a worst case maximum likelihood adversary

Statistical Privacy

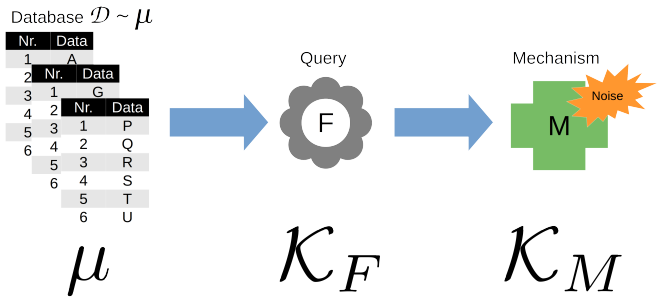
Model Limited Background Knowledge



Stochastic Model



Stochastic Model



Optimal δ

$$\hat{\Delta}^{M\circ F}_{\mathcal{D},j}(\varepsilon) := \max_{x,y} \underbrace{\int \max\{0, \mu_x \mathcal{K}_F \mathcal{K}_M(a) - e^{\varepsilon} \mu_y \mathcal{K}_F \mathcal{K}_M(a)\} \mathrm{d}a}_{\Delta^{M\circ F,j}_{\mathcal{D}_x,\mathcal{D}_y}(\varepsilon)}$$

Sampling

Use Cases

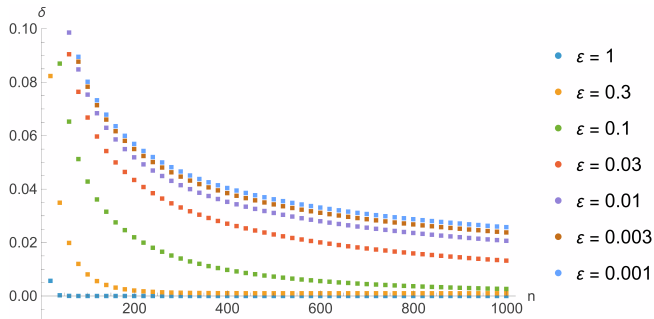
- common technique used in statistics
  - Bootstrapping
  - Cross validation
  - Jackknife
- alternative to additive noise
  - critical view of additive noise

Sampling

Use Cases

- common technique used in statistics
  - Bootstrapping
  - Cross validation
  - Jackknife
- alternative to additive noise
  - critical view of additive noise

Realtion between Database Size n and Privacy Parameters (ε, δ)

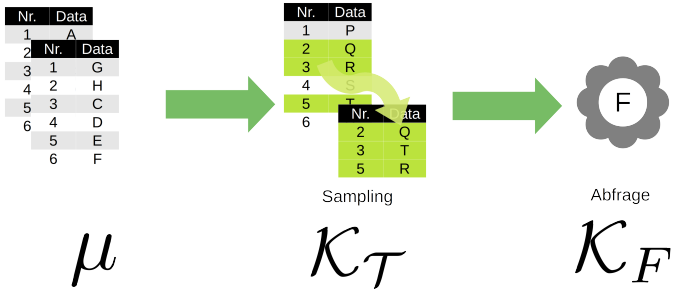


Sampling: Mathematical Basics

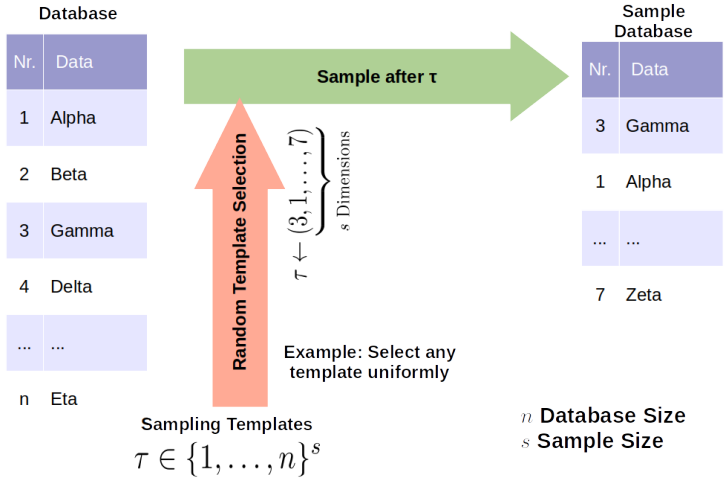
- entangling the database distribution and sampling
- applying advanced joint convexity [1]
- bounding the terms obtained
  - $\Delta^{M \circ F, j}_{\mathcal{D}_x, \mathcal{D}_y}(\varepsilon)$
  - $\Delta^{M \circ F, j}_{\mathcal{D}_x, \mathcal{D}}(\varepsilon)$

[1] Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences (2018), B. Balle, G. Barthe, M. Gaboardi

Sampling: Splitting the Densities



Structure of Sampling



Formal Modelling of Subsampling

- any sampling technique can be represented by the distribution of $\mathcal{T}$.
- $\mathcal{T}$ uniformly distributed on a subset of $\{1, \dots, n\}^s$

Reduce to fixed Samples

$$\mu\mathcal{K}_{\mathcal{T}}\mathcal{K}_F = \sum_{\tau} \mathbb{P}(\mathcal{T} = \tau)\mu\mathcal{K}_{\tau}\mathcal{K}_F$$

Modelling Structural Knowledge:

- exact values of a subset of entries
- exact number of adults and kids

$\Rightarrow$ modeled by a non-identical distributed database

Sampling Privacy Curve

$$\mathtt{SPC}^{F,\mathcal{T}}_{\mathcal{D},j}(\varepsilon) \quad := \quad \mathbb{E}_{\tau\sim\mathcal{T}}\left[\hat{\Delta}^{F\circ\mathtt{SAMP}_\tau}_{\mathcal{D},j}(\varepsilon)\right]$$

$\mathcal{D}$: database

$\mathcal{T}$: template distribution

j : sensitive entry

$\mathtt{SAMP}_\tau$: sample using τ

Bounding SPC

Assumptions

- symmetric query
- entries are drawn equally likely
  - minimizing statistical bias
- t types of entry distributions

Bounding

- hypergeometric, multivariate hypergeometric or multinomial distribution
- average over all database types
- bound by choosing a suitable threshold

Sampling Results

Sampling	ε	δ
w.o. rep.	$\log(1+(e^{\varepsilon}-1)\frac{s}{n})$	$\frac{s}{n}\ \mathrm{SPC}_{\mathcal{D},j}^{F,\mathcal{T}[n,s]}(\varepsilon)$
Poisson$_\lambda$	$\log(1+\lambda(e^{\varepsilon}-1))$	$\sum_{s=0}^{n}\binom{n}{s}\lambda^s(1-\lambda)^{n-s}\ \frac{s}{n}\ \mathrm{SPC}_{\mathcal{D},j}^{F,\mathcal{T}[n,s]}(\varepsilon)$

F : symmetric query

λ : Pois. parameter

j : sensitive entry

$\mathcal{D}$: database of size n

$\mathcal{T}[n,s]$: temp. dist. for w.o. rep.

s : sample size

Sampling Results for IID Databases

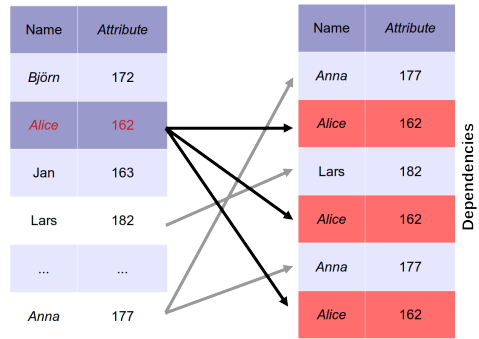
Sampling	ε	δ
w.o. rep.	$\log(1+(e^{\varepsilon}-1)\frac{s}{n})$	$\frac{s}{n}\ \hat{\Delta}^{F}_{\mathcal{D}_s,j}(\varepsilon)$
Poisson$_\lambda$	$\log(1+\lambda(e^{\varepsilon}-1))$	$\sum_{s=0}^{n}\binom{n}{s}\lambda^s(1-\lambda)^{n-s}\ \frac{s}{n}\ \hat{\Delta}^{F}_{\mathcal{D}_s,j}(\varepsilon)$

- F : symmetric query
- λ : Pois. parameter
- j : sensitive entry
- n : database size
- $\mathcal{D}_s$: database of size n
- s : sample size

Sampling With Replacement

Challenge: Multiple Draws

- the sensitive entry can be drawn multiple times
- drawing an entry multiple times introduces dependence
- fixing multiple entries can complicate the answer distribution
- determining the worst case is more complex



Sampling With Replacement

Nessisary Restrictions

- monotone queries
- symmetric queries
- $\Rightarrow$ controlling the influence of the most distinguishable entries

F-Samplable

- enforces separability of answer distributions for any sampling.
- $\frac{\mu_x \mathcal{K}_\tau \mathcal{K}_F}{\mu_y \mathcal{K}_\tau \mathcal{K}_F}$ is monotone for all x, y, τ.

Sampling with Replacement

ε	δ
$\log\left(1+(1-(1-\frac{1}{n})^s)(e^\varepsilon-1)\right)$	$\frac{n}{s}\sum_{\ell=1}^{s}\binom{n}{\ell}(\frac{1}{n})^\ell(1-\frac{1}{n})^{s-\ell}\,\mathtt{SPC}_{\mathcal{D},j}^{F,\mathcal{T}^\ell}(\varepsilon)$

F : symmetric query

j : sensitive entry

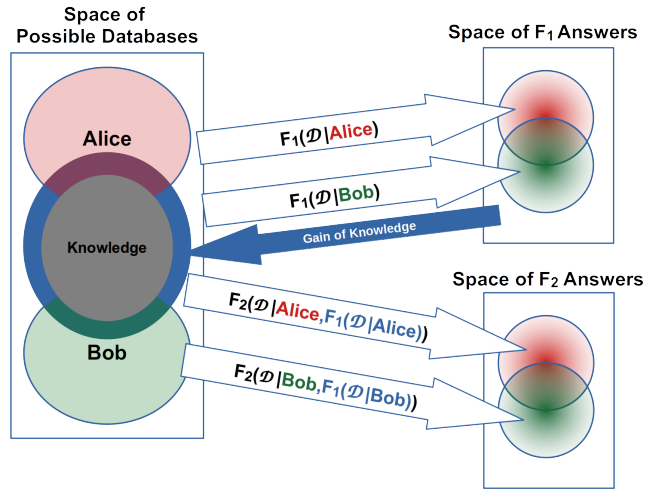
$\mathcal{D}$: F-samplable database of size n

$\mathcal{T}^\ell$: temp. dist. of size s where j is drawn ℓ times

The Composition Problem

Dependend Queries

- every queries reveals information on the database
- changing the underlying database distribution
- example: Number of people with condition A yesterday and today reveals information about a new person

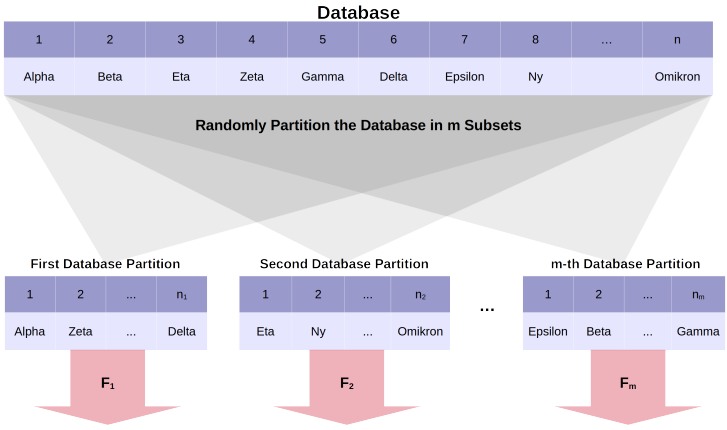


The Composition Problem

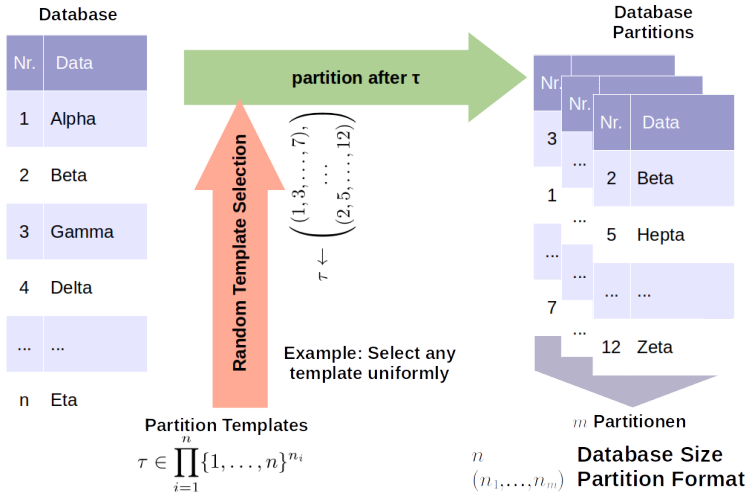
- determine sensible queries restrictions
  - symmetric queries
- reducing dependencies between queries
- bounding adversarial knowledge gain
- consider the privacy properties of each query

$$\Rightarrow \text{divide data among queries}$$

Random Allocation



Structure of Partitioning

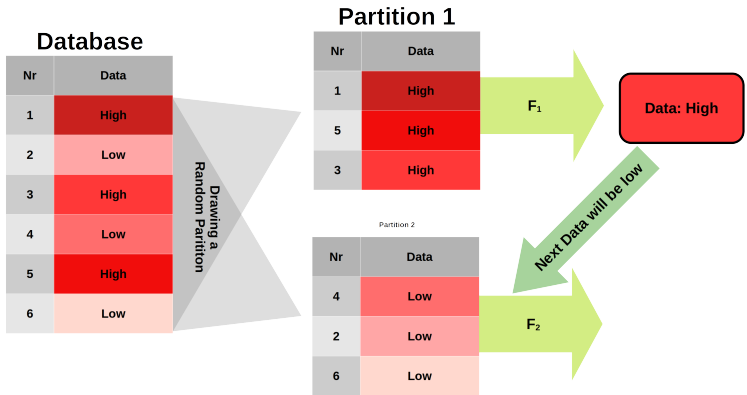


Random Partition

A random partition can be modeled by a uniformly distributed random variable $\mathcal{R} \sim \mu_{\mathcal{R}}$.

- $\mathcal{R}^k$ projection onto the k-th sample
- $\mathcal{R}_{j,k}$ restricting the j-th index to lie in the k-th sample
- $\mathcal{R}$ random partition of a database in parts of size $n_1, \dots, n_m$
- $\mathcal{R}^k_+$ distribution of the k-th partition when it includes the sensitive element

Partition Depencencies



Composition of Queries $F_1, \dots, F_m$

- F_k is a A_k valued subquery
- $F = (F_1, \dots, F_m)$ a composition
- non-adaptive: $F_k : D \mapsto A_k$
- adaptive: Defined by adversarial strategy $S(F_1, a_1, \dots, F_{k-1}, a_{k-1}) \mapsto F_k$:
  - F_k^S is not fixed but depends on previous answers
  - $F_k^S : \prod_{\ell=1}^{k-1} A_\ell \times D \mapsto A_k$

Composition for iid databases

no structural information can be leaked

δ given ε of composed query F:

	δ
Non Adaptive:	$\sum_{k=1}^{m} \frac{n_k}{n} \hat{\Delta}^{F_k}_{\mathcal{D}^{n_k},j}(\varepsilon)$
Adaptive:	$\sum_{k=1}^{m} \frac{n_k}{n} \mathbb{E}_{X\sim(F_1^S,\ldots,F_{k-1}^S)} \left[\hat{\Delta}^{F_k^S(X)}_{\mathcal{D}^{n_k}}(\varepsilon)\right]$

$\mathcal{D}^{n_k}$: database of size n_k

Composition Theorem

δ given ε of composed query F:

	δ
Non Adaptive:	$\sum_{k=1}^{m} \frac{n_k}{n} \ \mathrm{SPC}_{\mathcal{D},j}^{F_k,\mathcal{R}_+^k}(\varepsilon)$
Adaptive:	$\sum_{k=1}^{m} \frac{n_k}{n} \sum_{\tau} \mathcal{T}_{j\mapsto k}(\tau) \int_{A^{k-1}} \hat{\Delta}_{\mathtt{SAMP}_{\tau^k}(\mathcal{D})}^{F_k^S(\mathfrak{a}_{k-1})}(\varepsilon) \ \prod_{\ell<k} \mu_{F_\ell^S(\mathfrak{a}_{\ell-1}),\tau^k}(a_\ell) \ \mathbf{d}\mathfrak{a}_{k-1}$

$\mathcal{T}_{j\mapsto k}(\tau)$: temp. dist. where j is drawn for query k

$\mu_{F_\ell(\mathfrak{a}_{\ell-1}),\tau^k}(a_\ell)$: answer dist. of $F_\ell(\mathfrak{a}_{\ell-1})$ sampled by τ^k

$\mathfrak{a}_{\ell-1}$: the vector $(a_1, \dots, a_{\ell-1})$

Composition Example

Big Database

Using independent property queries with $p = 0.5$ and a database of size $n = 2^{15} = 32768$. Comparison with DP using Gaussian noise on queries without random allocation and equal accuracy. [2]

		$\varepsilon = 0.005$		$\varepsilon = 0.01$		$\varepsilon = 0.02$	
#SP	σ	δ	#DP	δ	#DP	δ	#DP
32	.0153	.0225	0	.0203	3	.0163	9
64	.0219	.0329	1	.0306	6	.0264	20
128	.0311	.0475	3	.0452	12	.0409	44
256	.0441	.0682	6	.0660	25	.0617	93
512	.0624	.0973	12	.0953	50	.0912	194

[2] The Composition Theorem for Differential Privacy (2017),P. Kairouz, S. Oh, P. Viswanath

Composition Example

Small Database

Using independent property queries with $p = 0.5$ and a database of size $n = 2^{10} = 1024$.

		$\varepsilon = 0.05$		$\varepsilon = 0.1$		$\varepsilon = 0.2$	
#SP	σ	δ	#DP	δ	#DP	δ	#DP
32	.0869	.1214	2	.1020	7	.0.0711	23
64	.1240	.1808	4	.1644	16	.1291	55
128	.1760	.2618	9	.2496	35	.2232	126

Future Work

- more general bounds for sampling with replacement
- comparison to additive noise
  - general model for additive noise interaction
- determining a way to bound adversarial knowledge in composition
- comparison randomness complexity of different mechanisms