Approximating the Optimal Composition of Differentially Private Mechanisms

Jack Murtagh* and Salil Vadhan**

*Research Assistant, Harvard School of Engineering and Applied Sciences

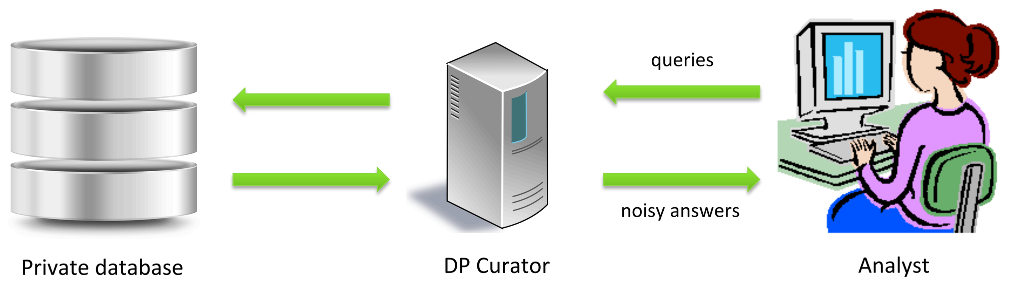
**Professor of Computer Science and Applied Mathematics, Harvard School of Engineering and Applied Sciences

Privacy Tools for Sharing Research Data

A National Science Foundation Secure and Trustworthy Cyberspace Project

Differential Privacy

- Promising avenue for answering statistical queries on sensitive datasets while minimizing privacy risks for individuals in the dataset



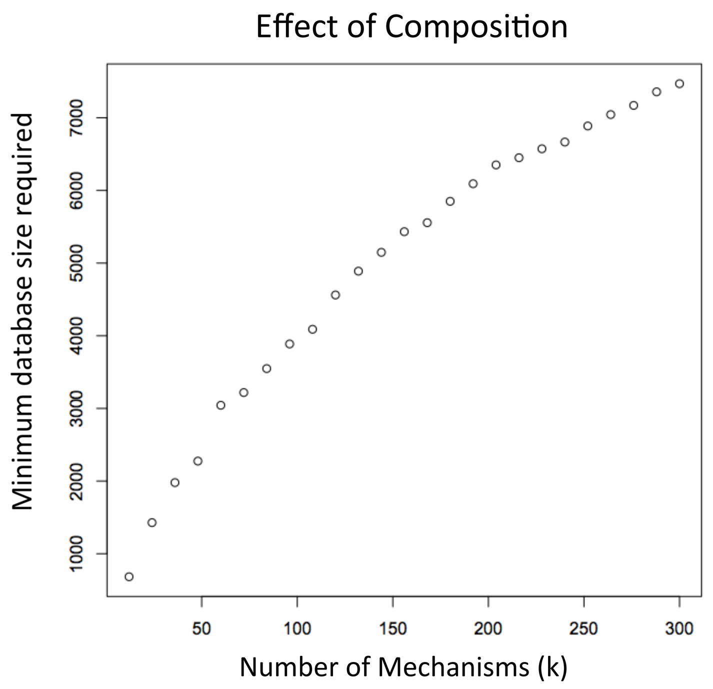
- An algorithm M is said to be (ε,δ)-Differentially Private if for all neighboring databases D, D' and all output sets S:

$$\Pr[M(D)\in S] \leq e^{\varepsilon} \Pr[M(D')\in S] + \delta$$

- Central Idea: A private algorithm should behave very similarly on a database with your data as it would on the **same** database **without** your data. This way, an adversary cannot distinguish the two cases.

Composition

- The more DP algorithms we run on a single database, the more privacy degrades



- Simple composition: If we run k (ε,δ)-DP mechanisms on the same database, we maintain a global privacy of (kε, kδ)
- But we want to compute many statistics on our database!
- Luckily, we can do better. Quantifying privacy degradation under composition as precisely as possible is of particular interest in the literature

Optimal Composition: Special Case

- Recent result [2] characterized optimal composition exactly:

Theorem 3.3. *For any $\varepsilon \geq 0$ and $\delta \in [0,1]$, the class of (ε,δ)-differentially private mechanisms satisfies*

$$\left((k-2i)\varepsilon\,,\,1-(1-\delta)^k(1-\delta_i)\right)\text{-differential privacy} \tag{5}$$

under k-fold adaptive composition, for all $i = \{0,1,\ldots,\lfloor k/2 \rfloor\}$, where

$$\delta_i \;=\; \frac{\sum_{\ell=0}^{i-1}\binom{k}{\ell}\left(e^{(k-\ell)\varepsilon}-e^{(k-2i+\ell)\varepsilon}\right)}{(1+e^{\varepsilon})^k}\,. \tag{6}$$

- Problem: This result only applies in the 'homogenous' case i.e. when every mechanism in the composition has identical privacy parameters.
- In practice, we often want to run algorithms with different privacy parameters on the same database

Optimal Composition: General Case

Question: Can we get a similar result for the general heterogeneous case?
Answer: Probably not

- Conjecture: computing the optimal composition for a general set of private mechanisms is #P-complete (so an efficient solution would at least imply P=NP)

$$\frac{\delta_g - 1 + \prod_{i=1}^k (1-\delta_i)}{\prod_{i=1}^k (1-\delta_i)} = \frac{1}{\prod_{i=1}^k (1+e^{\epsilon_i})} \sum_{x\in S} \frac{e^{\sum_{i=1}^k \epsilon_i}}{e^{<x,\bar{\epsilon}>}} - e^{\epsilon_g + <x,\bar{\epsilon}>}$$

where:

$$S = \{x \in \{0,1\}^k \mid \sum_{i=1}^{k} x_i\epsilon_i \leq \frac{\sum_{i=1}^k \epsilon_i - \epsilon_g}{2}\}$$

- Problem turns out to be very closely related to partition and knapsack-type problems, known to be #P-complete:

Given a set of integers $S = \{w_1, w_2, \ldots w_k\}$, how many ways can we partition S into disjoint P_1, P_2 (with $P_1 \cup P_2 = S$) such that: $\prod_{i\in P_1} w_i = \prod_{i\in P_2} w_i$

Approximation Algorithm

- Idea: If we can't compute optimal composition exactly, maybe we can approximate it
- There exist polynomial time algorithms for approximately counting knapsack solutions [1]
- Task: Modify a counting algorithm to sum knapsack solutions rather than count them and to let us get arbitrarily close to the optimal ε_g

Algorithm

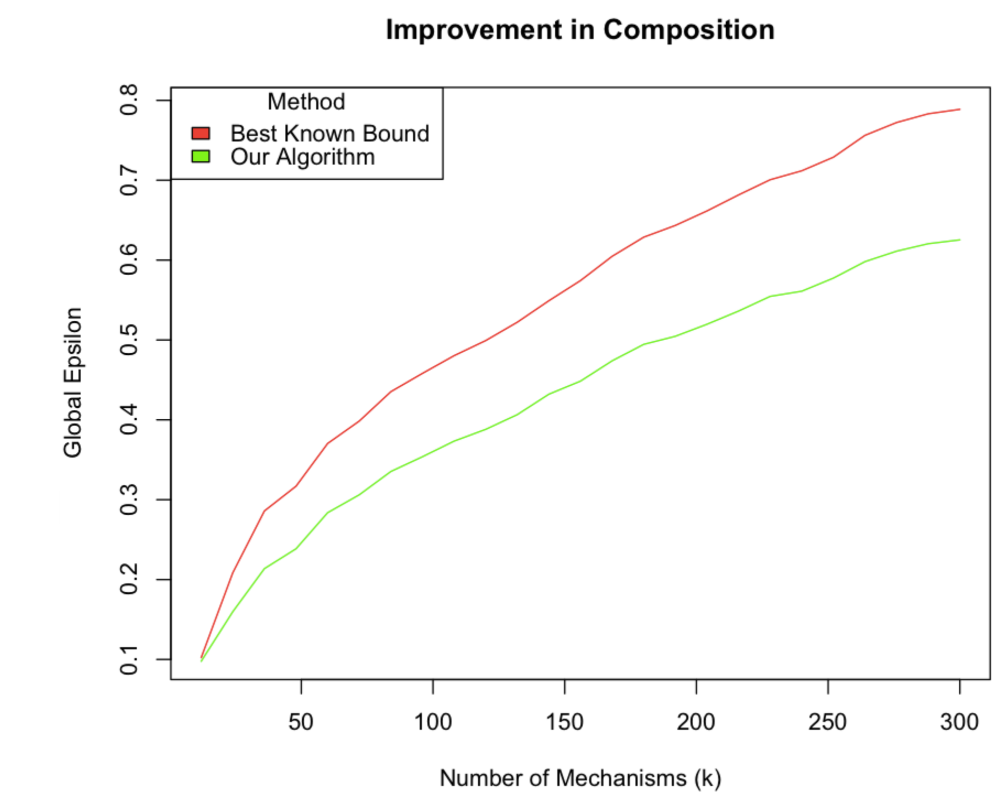
Inputs: $\varepsilon_1, \varepsilon_2 \ldots \varepsilon_k, \delta_1, \delta_2, \ldots, \delta_k, \delta_g, \varepsilon^*, t$
Outputs: True if $\varepsilon_g \leq \varepsilon^*$,
False if $\varepsilon_g > \varepsilon^* + O(k/t)$

1. Set $b = (\Sigma\varepsilon_i - \varepsilon^*)/2$
2. Set $a_i = \text{floor}(t\varepsilon_i/b)$
3. Set $f_i = ba_i/t$
4. Build k×t table using the recursion:
   $T(r,s) = T(r-1,s) + e^{f_i}\,T(r-1,s-a_i)$
   with $T(1,s) = e^{f_1}+1$ if $a_1 \leq s$
   and $T(1,s) = 1$ otherwise
5. Plug T(k, t) into optimal composition equation
6. If privacy satisfied, output: True, else: False

- Algorithm shifts ε's to a slightly different input (f's) that can be solved exactly with dynamic programming.
- Proved that this "shifted input" can change the answer by at most O(k/t)
- Runs in time O(kt) so can efficiently get arbitrarily close to true answer by using binary search on ε^*

Conclusions

- If computing optimal composition is #P-complete, an algorithm that approximates it to arbitrary precision in polynomial time is essentially the best we can hope for.
- Algorithm outperforms bound provided in [2] in practice:



Future Work

- Prove hardness result
- Try to improve running time of approximation - random sampling
- Integrate approximation algorithm with composition piece of privacy tools project

References

[1] Martin Dyer. Approximate counting by dynamic programming. Proceedings of the thirty-fifth annual ACM symposium on Theory of computing. ACM, 2003.

[2] Sewoong Oh and Pramod Viswanath. The composition theorem for differential privacy. arXiv preprint arXiv: 1311.0776 (2013).

Contact

Jack Murtagh
murtagh.jack@gmail.com

CRCS Center for Research on Computation and Society at Harvard School of Engineering and Applied Sciences

BERKMAN CENTER FOR INTERNET & SOCIETY AT HARVARD UNIVERSITY

The Institute *for* Quantitative Social Science at Harvard University

DATA PRIVACY LAB