CS229r: Mathematical Approaches to Data Privacy
Homework 2

Due Friday, March 15th, 2013 by 5PM.
(Email to `diffprivcourse-hw@seas.harvard.edu`)

- You must type your solutions. LaTeX or Microsoft Word are both fine. Let us know if there is something else you prefer to use. If you use LaTeX, please submit both the compiled PDF file and the source.
- Strive for clarity and conciseness in your solutions, emphasizing the main ideas over low-level details.
- Remember to cite all collaborators and sources of ideas.

Problem 1 (Asymptotic Optimality of Advanced Composition)**.** Recall that the advanced composition theorem says that the k-fold composition of an ε-differentially private mechanism is $(1, \delta)$ differentially private when $k \leq 1/(8\varepsilon^2 \log(1/\delta))$. Show that for some choice of $k = \tilde{O}(1/\varepsilon^2)$, the k-fold composition of the ε-differentially private Laplace mechanism is *not* $(1, 1/10)$ differentially private. ($\tilde{O}(x)$ means an expression that can be bounded by $x \cdot \text{polylog}(x)$; the polylogs are actually not necessary for this problem, but might come up depending on how you solve the problem.)

Problem 2 (Differentially Private Global Minimum Cut)**.** Let $G = (V, E)$ be an undirected, unweighted graph. For every $S \subseteq V$, define $E(S, S^c) = |\{(u, v) \in E \mid u \in S, v \notin S\}|$ to be the number of edges "cut by S". The *global min-cut* problem is to find the set S that cuts the fewest edges. That is, to find

$$S_{OPT} = \operatorname*{argmin}_{S \subseteq V} |E(S, S^c)|.$$

Let $OPT = |E(S_{OPT}, S^c_{OPT})|$ be the number of edges crossing the global min-cut.

1. Give an instantiation of the exponential mechanism that will output an ε-differentially private set $\widehat{S}$ that is an approximate global minimum cut in G. Here we refer to *edge-level* differential privacy, meaning that removing or adding one edge to the graph should not change the output distribution of the mechanism by more than an e^{ε} factor.

2. A nice property of global minimum cuts is that there cannot be too many nearly-minimum cuts. In particular, it is known that, for every $C > 0$, the number of cuts S such that $|E(S, S^c)| \leq C \cdot OPT$ is at most $|V|^{2C}$. Use this fact to show that, when $OPT \geq 8 \log(|V|/\beta)/\varepsilon$, with probability at least $1 - \beta$, your mechanism will output a set $\widehat{S}$ such that

$$|E(\widehat{S}, \widehat{S}^c)| \leq OPT + O\left(\frac{\log(|V|/\beta)}{\varepsilon}\right)$$

   So as long as the global min-cut in G doesn't cut too few edges, the exponential mechanism will privately find a nearly-minimum cut.

Problem 3 (Synthetic Data)**.** Recall that, given a family of linear queries $\mathcal{C} = \{f_1, \ldots, f_k\}$, and a database $x \in \mathcal{X}^n$, an *α-accurate synthetic database for x and $\mathcal{C}$* is a database $\widehat{x} \in \mathcal{X}^{\widehat{n}}$ such that

$$\forall f \in \mathcal{C} \, |f(x) - f(\widehat{x})| \leq \alpha$$

(Here, as in Chapter 4 of Dwork–Roth, a linear query f is specified by a function $f : \mathcal{X} \to [0,1]$, and we extend it to databases $x = (x_1, \ldots, x_n)$ by averaging: $f(x) = (1/n) \cdot \sum_i f(x_i)$. Equivalently, we view the database as a distribution $x \in [0,1]^{\mathcal{X}}$ (where $x(j)$ is the fraction of database elements of type j) and then $f(x) = \sum_j f(j)x(j)$.)

In this problem, you will show that every set of queries that can be answered privately and accurately, can also be answered privately and accurately with a synthetic database. More precisely, suppose that $M : \mathcal{X}^n \to \mathbb{R}^{|\mathcal{C}|}$ is (ε, δ)-dp and answers every $f \in \mathcal{C}$ up to error $\pm\alpha$, with probability at least $1 - \beta$. That is, with probability at least $1 - \beta$,

$$\forall f_i \in \mathcal{C}, |f_i(x) - M_i(x)| \leq \alpha.$$

1. Show that there exists an (ε, δ)-dp mechanism $M' : \mathcal{X}^n \to \mathcal{X}^{\widehat{n}}$, that with probability at least $1 - \beta$ outputs an $O(\alpha)$-accurate synthetic database for x and $\mathcal{C}$. (You can choose the value of $\widehat{n}$.)
2. Show how to obtain such a mechanism M' that runs in time $T_M + \text{poly}(|\mathcal{X}|, k)$, where T_M is the running time of M. (**Hint:** Use linear programming to find a suitable distribution over the data universe and then sample from that distribution to generate the accurate synthetic database.)
3. **Challenge Problem:** Use a similar technique as Part 2 to design a computationally efficient analogue of Dwork–Roth Theorem 108, for the case that all queries are answered accurately. That is, exhibit a (poly(n)-time adversary that reconstructs all but $o(n)$ entries of a database $d \in \{\pm 1\}^n$ with high probability when given $m = O(n)$ random ± 1 vectors $v_1, \ldots, v_m \in \{\pm 1\}^n$ and answers $y_1, \ldots, y_m \in \mathbb{R}$ such that $|y_i - \langle d, v_i \rangle| \leq o(\sqrt{n})$ for all $i = 1, \ldots, m$. (Note that the order of database rows matters in this problem.)

Problem 4. One natural type of data analysis is to check determine whether almost all of the individuals in a dataset share some attribute (e.g. their DNA all agrees on a particular gene) and if so output that attribute. That is, we want a mechanism $M : \mathcal{X}^n \to \mathcal{X}$ with the following properties:

- If at least 90% of the rows in database x are all equal to each other, say having value v, then $\Pr[M(x) = v] \geq 2/3$.
- If database x does not have 80% or more of its rows all equal to each other, then $\Pr[M(x) = \perp] \geq 2/3$.

Note that there are natural cases of this problem where the data universe $\mathcal{X}$ is very large (e.g. consisting of all alleles of particular gene).

1. Show that if such a mechanism M is $(\varepsilon, 0)$-differentially private, then $n = \Omega(\log |\mathcal{X}|/\varepsilon)$.
2. On the other hand, show that there is an (ε, δ)-differentially private mechanism that works for some choice of $n = O((1/\varepsilon) \cdot \log(1/\delta))$; note that there is no dependence on $|\mathcal{X}|$. (Hint: first estimate the frequency of the most frequent item in the database, and act differently based on whether this frequency is large or not.)