ISYE 8813: Algorithmic Foundations of Ethical ML **Fall 2021**

Lectures 9-10: SmallDB

Lecturer: Juba Ziani

So far we have seen tools (Laplace, Exponential, Randomize Response) that allowed us to answer k queries with noise that scaled like $\Theta(k)$ for $(\epsilon, 0)$-DP, and noise that scaled like $\Theta(\sqrt{k \ln(1/\delta)})$ for (ϵ, δ)-DP. Then we saw Sparse Vector, which allowed our noise to scale like $\Theta(\log k)$, with the caveat that we were only allowed to provide answers to $c \ll k$ queries, i.e. only those with "interesting"/outlying values. However, up until now, we were assuming all the queries were independent of each other, and adding independent noise to them.

Today, we study the SmallDB Mechanism, which allows us to actually answer all k queries, while only adding noise that scales like $\Theta(\log k)$. The key trick to making that work is to *correlate* the noise we add across queries. For example, imagine you first ask the mechanism to answer query f, and it outputs $f(x) + Lap(\frac{\Delta f}{\epsilon})$, and you then ask the mechanism to answer the exact same query f again. It should not redraw fresh noise to answer the query, but should instead tell you that you already know the appropriate answer. SmallDB does this by simply outputting a small database which provides approximately accurate answers to all queries in a pre-specified class with high probability. It uses the Exponential Mechanism to select such a database.

We note that this is a simple example of DP mechanisms outputting *synthetic data*, rather than directly outputting answers to said queries on the true data. Note that releasing such synthetic data can be very useful in practice: if we have a dataset composed of sensitive and private data points that we may not want to share with a third-party, for privacy reason, but we want to allow the third-party to still be able to perform his or her own statistical analysis on this data, we can instead release the sanitized/private synthetic data to them. For example, many research papers rely on the use of publicly available data; with synthetic data, we can now publicly release a differentially private version of this data with similar statistic properties as the original dataset, even if this dataset contains private data and cannot be released itself.

1 SmallDB

Before we see the mechanism, let us formally define the problem.

1.1 Query Release Problem

Definition 1 (Query Release Problem)**.** *In the* query release problem*, given a class of queries Q, the goal is to release an answer a_i to each query $f_i \in Q$ such that the worst-case additive error $\max_i |a_i - f_i(x)|$ is small, and our method for producing these answers $\{a_i\}$ satisfies differential privacy.*

We will typically consider the class Q of normalized linear queries, as defined below. We note that the normalization refers to ensuring the query has range $[0, 1]$. Any linear query with bounded range can be normalized to produce answers in this range.

Definition 2. *A (normalized) linear query f over a data universe $\mathcal{X}$ is of the form $f : \mathcal{X} \to [0, 1]$, where the query assigns a numerical value to every element of the data universe. To apply the linear query to a database $x = (x_1, ..., x_{|\mathcal{X}|}) \in \mathbb{N}^{|\mathcal{X}|}$ (i.e., in histogram notation), we abuse notation and define $f(x)$ to be the average value of the query f on the database. That is,*

$$f(x) = \frac{1}{\|x\|_1} \sum_{i=1}^{|\mathcal{X}|} x_i f(\mathcal{X}_i).$$

Informally, when a data point has value $\mathcal{X}_i$, query f has value $f(\mathcal{X}_i)$; in the histogram representation, this happens x_i times/for x_i data points, hence the total sum of the values present in database x is given by $\sum_{i=1}^{|\mathcal{X}|} x_i f(\mathcal{X}_i)$. Since we have that $\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} x_i = n$ is the number of data points in the database (since x_i is the number of data points of type $\mathcal{X}_i$), the average across all data points is given by $\frac{1}{\|x\|_1} \sum_{i=1}^{|\mathcal{X}|} x_i f(\mathcal{X}_i)$.

Note that $0 \leq f(x) \leq 1$ for any database $x \in \mathbb{N}^{|\mathcal{X}|}$, and this implies that the sensitivity is given by $\Delta f \leq 1$.

1.2 SmallDB Mechanism

SmallDB in Algorithm 1 takes in a database x, a class of queries Q, a privacy parameter ϵ, and an accuracy parameter α. It outputs a database y, whose size depends on the log of the number of queries you want to answer, and on the desired accuracy guarantee. It picks this database y by running the Exponential Mechanism with utility function of the negative error to the query release problem.

Importantly, SmallDB does not answer these queries, but rather, produces synthetic data from which you can compute the answers yourself. This is how it correlates noise across queries — not by explicitly correlating additive noise to query answers, which can be problematic, but by generating a shared data structure used to answer queries.

Algorithm 1: SmallDB (x, Q, ϵ, α)

Input: database x, query class Q, privacy parameter ϵ, accuracy parameter α
Let $\mathcal{R} = \{y \in \mathbb{N}^{|\mathcal{X}|} : \|y\|_1 = \frac{\log |Q|}{\alpha^2}\}$.
Let $u : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \to \mathbb{R}$ be:

$$u(x, y) = -\max_{f \in Q} |f(x) - f(y)|.$$

Sample and output $y \in \mathcal{R}$ with the Exponential Mechanism $M_E(x, u, \mathcal{R}, \epsilon)$.

Privacy of SmallDB: The privacy guarantee of SmallDB is immediate.

Theorem 3. *SmallDB is* $(\epsilon, 0)$*-DP.*

Proof. SmallDP is an instantiation of the Exponential Mechanism. Privacy of SmallDB hence follows immediately from privacy of the Exponential Mechanism. □

Accuracy of SmallDB: Accuracy of SmallDB is a bit trickier. We will proceed by showing:

1. There exists a "good" small database, in the sense that is approximately correct/accurate on all queries, compared to if we were using the true database x. That is, there exists a database y with $\|y\|_1 = \frac{\log|Q|}{\alpha^2}$ such that $-u(x,y) = \max_{f \in Q} |f(x) - f(y)| < \alpha$.
2. We will sample such a "good" database with high probability.

We need both of these steps because the Exponential Mechanism's accuracy guarantees only ensure that with high probability we will sample an output that has quality score (i.e., query release error) close to that of the optimal *small* database; this is Step 2.

However, in our setting, we do not know the magnitude of the error of the optimal small database compared to the original database x. This is what Step 1 gives us, and this why we need both parts to bound accuracy of the SmallDB algorithm.

1.3 Chernoff bounds

First, we are going to digress and talk about Chernoff bounds, which we will need for the proof of SmallDB accuracy. Chernoff bounds are an example of concentration inequalities (much like Chebyshev and Markov that we used earlier in the class), that bounds how far the empirical mean of several sampled random variables is from the expectation of the distribution these variables are from, with high probability. The advantage of using Chernoff bounds is that they show an exponentially decrease in the probability of being far away from the mean as we obtain more samples; this is a much faster decrease/convergence rate than what we saw with Chebyshev and Markov.

Theorem 4. *Let* $X_1, ..., X_n$ *be independent random variables bounded such that* $0 \leq X_i \leq 1$ *for all* $i \in [n]$*. Let* $S = \frac{1}{n}\sum_{i=1}^{n} X_i$ *denote their sample mean, and let* $\mu = \mathbb{E}[S]$ *denote their expected mean. Then, (additive Chernoff bounds),*

$$\Pr[S > \mu + \alpha] \leq e^{-2n\alpha^2},$$

$$\Pr[S < \mu - \alpha] \leq e^{-2n\alpha^2},$$

and (multiplicative Chernoff bounds),

$$\Pr[S > (1+\alpha)\mu] \leq e^{-n\mu\alpha^2/3},$$

$$\Pr[S > (1-\alpha)\mu] \leq e^{-n\mu\alpha^2/2}.$$

1.4 Step 1: There exists a good small database

Back to accuracy of SmallDB, armed with Chernoff bounds, we will start with step 1 above, by showing that there exists a "good" small database.

Theorem 5. *For any finite class of linear queries Q and any $\alpha > 0$, if $\mathcal{R} = \{y \in \mathbb{N}^{|\mathcal{X}|} : \|y\|_1 = \frac{\log|Q|}{\alpha^2}\}$, then for all $x \in \mathbb{N}^{|\mathcal{X}|}$, there exists $y \in \mathcal{R}$ such that*

$$\max_{f \in Q} |f(x) - f(y)| \leq \alpha.$$

Note that this is not saying anything about the SmallDB algorithm of which y is selected by the algorithm, but rather the existence of such a good small database in the set $\mathcal{R}$.

Proof. We will construct such a database y by taking $m = \frac{\log|Q|}{\alpha^2}$ samples uniformly at random (with replacement) from the elements of x. Let $m = \frac{\log|Q|}{\alpha^2}$ and let $s_1, \ldots, s_m$ be sampled i.i.d. from the following distribution:

$$\Pr[s_i = \mathcal{X}_j] = \frac{x_j}{\|x\|_1} \quad \forall i \in [m] \text{ and } \forall j \in [|\mathcal{X}|].$$

Define database y to contain the elements $s_1, \ldots, s_m$. For any $f \in Q$, we have

$$f(y) = \frac{1}{\|y\|_1} \sum_{i=1}^{|\mathcal{X}|} y_i f(\mathcal{X}_i) = \frac{1}{m} \sum_{i=1}^{m} f(s_i),$$

where the first equality is by the definition of linear queries, and that the second is obtained by switching to look at value per entry in y. That is, we can view the function value $f(y)$ equivalently under the histogram database notation, or the matrix/multiset of row database notation.

We are now looking at an average of independent random variables bounded between $0 \leq f(s_i) \leq 1$, so we can use our new tool of Chernoff bounds to show that the empirical mean of the $f(s_i)$ is close to its expectation $\mathbb{E}[f(s)]$. Then, because we take $s = (s_1, \ldots, s_m)$ be from the same distribution as the empirical distribution/histogram of data points in database x, the mean of the linear query on s will be the same as the mean of the query on x.

$$\begin{aligned}
\mathbb{E}[f(y)] = \mathbb{E}[\frac{1}{m} \sum_{i=1}^{m} f(s_i)] &= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}[f(s_i)] \\
&= \frac{1}{m} \sum_{i=1}^{m} \left(\sum_{j=1}^{|\mathcal{X}|} \frac{x_j}{\|x\|_1} f(\mathcal{X}_j) \right) \\
&\overset{*}{=} \frac{1}{m} \sum_{i=1}^{m} f(x) \\
&= f(x)
\end{aligned}$$

Applying an additive Chernoff bound, we get,

$$\Pr[|f(y) - f(x)| > \alpha] \leq 2e^{-2m\alpha^2}.$$

Taking a union bound over all linear queries of $f \in Q$ gives that

$$\Pr[\max_{f \in Q} |f(y) - f(x)| > \alpha] \leq 2|Q|e^{-2m\alpha^2} < 1,$$

for our choice of $m = \frac{\log |Q|}{\alpha^2}$. □

1.4.1 The probabilistic method

Why is our proof above complete? We have only shown that there is probability strictly less that 1 of sampling a small database that has error less than α. Why does this complete our proof?

We randomly sampled a database y of size $\frac{\log |Q|}{\alpha^2}$. Through this random sampling process, we found that with probability strictly less than 1, $f(y)$ will be more than α away from the desired answer $f(x)$ on some query f. Let's think of the reverse: if this probability was exactly 1, this would mean that *all* databases of size $\frac{\log |Q|}{\alpha^2}$ sampled from x would have some query f for which it has additive error greater than α.

Since the probability is instead *strictly less than 1*, it means that there is some database y we could have sampled that has $|f(y) - f(x)| < \alpha$ for all $f \in Q$. This does not tell us what that database is or how to find it, but this tells us such a database exists.

This is an example of proof by the probabilistic method: considering a random process and showing that there is a strictly positive probability of some good event happening. This tells you that some realization of the randomness in that random process caused that good event to happen, so there must exist *some* good realization for which your good event occurs.

Back to the proof, we have shown that there exists a good y of size $\frac{\log |Q|}{\alpha^2}$, and $\mathcal{R}$ contains all databases of size $\frac{\log |Q|}{\alpha^2}$, so it must contain at least one good database for every input x.

1.5 Step 2: Selecting a good database

It now remains to prove that we can sample such a "good" database with high probability. We will use the accuracy theorem of the Exponential Mechanism for this.

Proposition 6. *Let Q be a finite class of linear queries, and let $y = SmallDB(x, Q, \epsilon, \alpha)$. Then, with probability $\geq 1 - \beta$,*

$$\max_{f \in Q} |f(x) - f(y)| < \alpha + \frac{2\left(\frac{\log |\mathcal{X}| \cdot \log |Q|}{\alpha^2} + \log(1/\beta)\right)}{\epsilon \|x\|_1}.$$

Proof. Recall the accuracy theorem for Exponential Mechanism: for any $\beta > 0$ we have

$$\Pr[u(M_E(x, u, \mathcal{R}, \epsilon)) \leq OPT_u(x) - \frac{2\Delta u(\ln |\mathcal{R}| + \log(1/\beta))}{\epsilon}] \leq \beta. \quad (1)$$

We will instantiate this theorem with the relevant parameters for SmallDB: (1) by construction, $|\mathcal{R}| = |\mathcal{X}|^{\frac{\log|Q|}{\alpha^2}}$, so $\ln|\mathcal{R}| = \frac{\log|\mathcal{X}|\log|Q|}{\alpha^2}$, (2) by definition of the utility function, $u(M_E(x,u,\mathcal{R},\epsilon)) = u(y) = \max_{f\in Q}|f(y)-f(x)|$, (3) by Theorem 5, $OPT_u(x) \leq \alpha$, and (4) $\Delta u = \frac{1}{\|x\|_1}$.

Plugging these parameter values into Equation (1) gives the desired bound:

$$\Pr\left[\max_{f\in Q}|f(x)-f(y)| \geq \alpha + \frac{2\left(\frac{\log|\mathcal{X}|\cdot\log|Q|}{\alpha^2} + \log(1/\beta)\right)}{\epsilon\|x\|_1}\right] \leq \beta.$$

□

1.6 Putting it all together

These two steps combine to give us our final SmallDB accuracy guarantee.

Theorem 7. *Let y be the database output by SmallDB$(x, Q, \epsilon, \alpha/2)$. Then with probability at least $1-\beta$,*

$$\max_{f\in Q}|f(y)-f(x)| \leq \left(\frac{16\log|\mathcal{X}|\log|Q| + 4\log(1/\beta)}{\epsilon\|x\|_1}\right)^{1/3}.$$

Proof. By Proposition 6, $y = \text{SmallDB}(x, Q, \epsilon, \alpha/2)$ satisfies,

$$\Pr\left[\max_{f\in Q}|f(x)-f(y)| \geq \alpha/2 + \frac{2\left(\frac{4\log|\mathcal{X}|\cdot\log|Q|}{\alpha^2} + \log(1/\beta)\right)}{\epsilon\|x\|_1}\right] < \beta.$$

Setting $\alpha/2 = \frac{2\left(\frac{4\log|\mathcal{X}|\cdot\log|Q|}{\alpha^2}+\log(1/\beta)\right)}{\epsilon\|x\|_1}$, give the optimized bound in the theorem statement. □

2 Improved SmallDB accuracy bounds using VC-dimension

We proved the previous result by showing that there exists a good database of size $\frac{\log|Q|}{\alpha^2}$, or equivalently that there is a small set of size at most $|\mathcal{X}|^{\frac{\log|Q|}{\alpha^2}}$ which must contain a good outcome. This dependence on $\log|Q|$ assumes nothing about the structure of the class Q, and in some cases, we can do better. For example, what if Q is just the same query over and over? What if Q is infinite, but is well approximated by finite databases (e.g., queries asking whether a point lies within a given interval of the real line)?

For this section, we are going to restrict to counting queries, $f : \mathcal{X} \to \{0,1\}$—a subclass of linear queries with binary outputs—and will improve the bound of Theorem 7 using VC-dimension, which is a measure of how complex a class of queries is. We are calling these queries as counting queries as $f(x) = \frac{1}{n}\sum_i x_i$ (in matrix/multiset of row representations) counts the number of entries with $f(x_i) = 1$.

2.1 VC-Dimension

Definition 8 (Shattering)**.** *A class of counting queries Q* shatters *a collection of points S if for every $T \subseteq S$, there exists an $f \in Q$ s.t. $\{x \in S | f(x) = 1\} = T$.*

That is, Q shatters S if for every one of the $2^{|S|}$ subsets T of S, there is some function in Q that labels exactly those elements as positive, and does not label any elements in $S \setminus T$ as positive. Intuitively, this means that *no matter how I label my points* (think of T being a labelling, where points in T are labelled positively and points not in T are labelled negatively), there is a query f in my class Q that *classifies these points perfectly.*

Example: We will consider some examples $S \subseteq \mathbb{R}^2$, and let Q be counting queries that define half-spaces (linear classifier) in $\mathbb{R}^2$. For each set, we will ask: does Q shatter S?

These examples are illustrated below, where we either show the collection of half-spaces that shatter S, or we show a set of points that cannot be exclusively labeled as positive by any half-space in $\mathbb{R}^2$.

1. S_1 Two points. Answer: Yes.
2. S_2 Three points that do not lie one the same line. Answer: Yes.
3. S_3 Three points lie on the same line. Answer: No.
4. S_4 Four points lie on a quadrilateral. Answer: No.

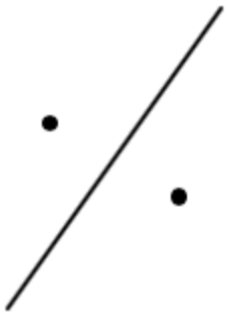

Figure 1: S_1

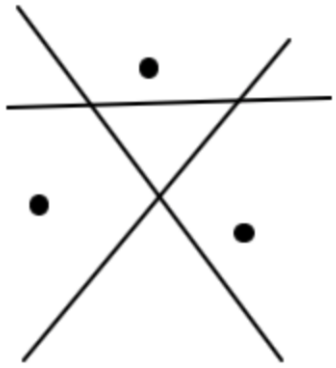

Figure 2: S_2

Definition 9 (Vapnik-Chervonenkis (VC) dimension)**.** *A collection of counting queries Q has* VC-Dimension *d if there exists some set $S \subseteq \mathcal{X}$ of cardinality $|S| = d$ such that Q shatters S, and Q does not shatter any set of cardinality $d + 1$. We denote this quantity VC-DIM(Q).*

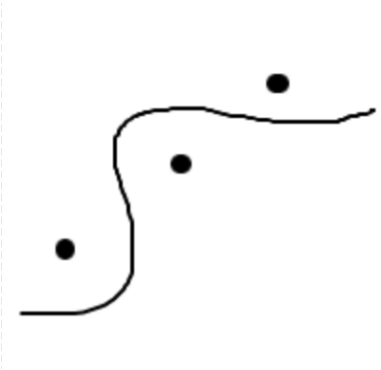

Figure 3: S_3

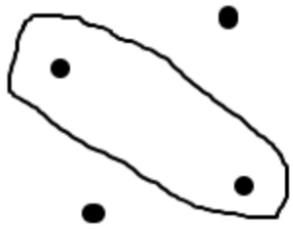

Figure 4: S_4

Returning to the example where Q is the set of all counting queries that define half-spaces in $\mathbb{R}^2$, then VC-DIM(Q)=3. We saw that Q shattered S_2 and $|S_2| = 3$. Also note that any set S with $|S| = 4$ must either have all four points on a quadrilateral as in S_4 (note that this includes the case in which one point is in the convex hull of the three remaining points), or three points on a line as in S_3, or have multiple co-located points. None of these cases can be shattered by Q.

The next lemma says that for any finite query class, the VC-dimension is not too large.

Lemma 10. *For any finite class Q, VC-DIM(Q) $\leq \log |Q|$.*

Proof. If VC-DIM(Q)=d, then Q shatters some set of items $S \subseteq \mathcal{X}$ with cardinality $|S| = d$. Then S must have 2^d distinct subsets, and $|Q| \geq 2^d$ since Q must contain a distinct function f for each subset of S. □

2.2 Better SmallDB bounds

Returning to our SmallDB bounds, we can plug in VC-DIM(Q) instead of $\log |Q|$, and by Lemma 10, this can improve the accuracy guarantee.

Theorem 11. *Let y be the database output by SmallDB$(x, Q, \epsilon, \alpha/2)$. Then with probability at least $1 - \beta$,*

$$\max_{f \in Q} |f(y) - f(x)| \leq O\left(\left(\frac{\log |\mathcal{X}| \cdot \textit{VC-DIM}(Q) + \log(1/\beta)}{\epsilon \|x\|_1}\right)^{1/3}\right).$$

Proof. We will not re-do the whole proof here. Rather, we will just re-do the first part, showing the existence of a small database y such that for all $f \in Q$, we have that

$$\sup_{f \in Q} |f(x) - f(y)| \leq \alpha,$$

and see how to optimally choose $\|y\|_1$. The rest of the proof will be the same as before, invoking the accuracy guarantee of the exponential mechanism. The only thing that will

change will be the $\ln|\mathcal{R}|$ terms, as a function of the range $\mathcal{R}$ we pick. To design $\mathcal{R}$, we will rely on the following theorem:

Theorem 12. *Let Q be a class of counting queries, and let $d = $ VC-DIM(Q). Let $S_1, \ldots, S_m$ be i.i.d. random variables. Then,*

$$\Pr\left[\sup_{f\in Q}\left|\mathbb{E}[f(S_i)] - \frac{1}{n}\sum_{i=1}^{n} f(S_i)\right| \geq \varepsilon\right] \leq Ce^d e^{-Km\varepsilon^2}$$

for some well-chosen constants C, K.

[Note that most classes/books show a weaker version of this theorem. For this version of the theorem, please refer to: M. Talagrand, "Sharper Bounds for Gaussian and Empirical Processes."]

One can see the above as an analog of the Chernoff bound we derived previously for the case in which Q was a finite query class. The theorem basically tells us that for any $f \in Q$, the empirical loss f when facing samples $S_1, \ldots, S_m$ concentrates around the expected loss $\mathbb{E}[f(S)]$. This theorem is often used in machine learning to show that empirical loss minimization works, in that picking the hypothesis in an hypothesis class Q with minimal empirical 0-1 classification loss also approximately minimizes the expected 0-1 loss (the probability of making a wrong prediction).

Now, suppose S_i's are i.i.d. samples taken uniformly at random from the rows of database x (i.e., $S_i = x_i$ with probability $1/n$), and let $y = (S_1, \ldots, S_m)$. Similarly to the finite query class case, we have that

$$\mathbb{E}[f(S_i)] = \frac{1}{n}\sum_{i=1}^{n} x_i = f(x),$$

as we select x_i with probability $1/n$. We also have by definition of $y = (S_1, \ldots, S_m)$ and the fact that f is a counting (hence linear) query that

$$f(y) = \frac{1}{n}\sum_{i=1}^{n} f(S_i).$$

We have therefore argued that

$$\Pr\left[\sup_{f\in Q}|f(x) - f(y)| \geq \alpha\right] \leq Ce^d e^{-Km\alpha^2}.$$

Now, plugging in $m = \frac{d+\ln C}{K\alpha^2}$ we get that

$$\Pr\left[\sup_{f\in Q}|f(x) - f(y)| \geq \alpha\right] \leq Ce^d e^{-d-\ln C)} = 1.$$

In turn, setting $m > \frac{d+\ln C}{K\alpha^2}$ – in particular, setting $m = \gamma\frac{d}{\alpha^2}$ for a big enough constant γ guarantees that the above probability is *strictly* less than 1. As before, this implies the existence of a database with $\|y\|_1 = \frac{\gamma d}{\alpha^2}$ with $\sup_{f\in Q}|f(x) - f(y)| \leq \alpha$.

Now, in that case, $\mathcal{R} = \{y : \ \|y\|_1 = \gamma d/\alpha^2\}$, and there are $|\mathcal{R}| = |\mathcal{X}|^{\gamma d/\alpha^2}$. This gives the desired

$$\ln |\mathcal{R}| = O\left(\frac{d \log |\mathcal{X}|}{\alpha^2}\right)$$

that we find in the final bound of Theorem 11. □