ISYE/CS 8803: Foundations of Data Privacy **Fall 2017**

Lecture 9: Private Multiplicative Weights

Lecturer: Rachel Cummings *Scribe: Wanrong Zhang, September 20, 2017*

1 Review of SmallDB

Theorem 1. *Let $y = SmallDB(x, Q, \epsilon, \alpha/2)$, then with probability $\geq 1 - \beta$,*

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

We proved this 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 $\mathcal{X}^{fraclog|Q|\alpha^2}$ which must contain a good outcome. This dependence on $log|Q|$ assumes nothing about the structure of the class Q, and sometimes 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.)

2 Improving SmallDB Bounds by VC Dimension

Today, we are going to restrict to counting queries: $f : \mathcal{X} \to 0, 1$ with Boolean outputs, and improve the bound of Theorem 1 using VC-dimension. For improved bounds for linear queries $f : \mathcal{X} \to [0, 1]$, we need fat shattering dimension (See Section 5.1 of the textbook [DR14], or paper [Rot10])

Definition 2. *A class of counting queries Q shatters a collection of points $S \subseteq \mathcal{X}$ 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.

Example: $S \subseteq \mathbb{R}^2$ and let Q be counting queries that define halfspaces in $\mathbb{R}^2$. Does Q shatter S?

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.

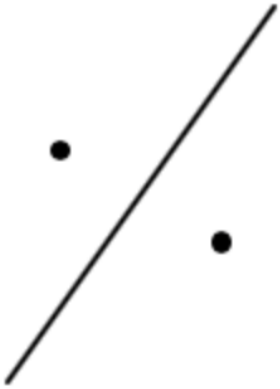

Figure 1: S1

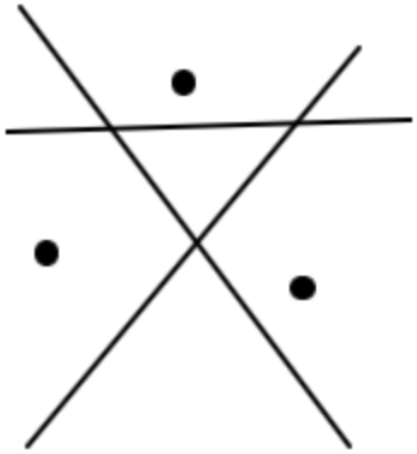

Figure 2: S2

Figure 3: S3

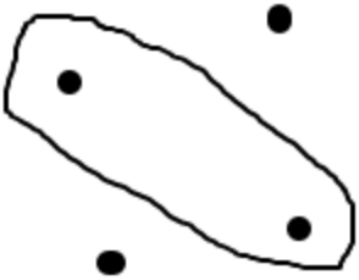

Figure 4: S4

Definition 3. *(Vapnik-Chervonenkis (VC) Dimension) A collection of counting queries Q has VC-dimension d if there exists some set $S \subseteq X$ of cardinality $|S| = d$ s.t. Q shatters S, and Q does not shatter any set of cardinality $d+1$. We denote this quantity $VC - DIM(Q)$.*

Returning to Example where Q is the set of all counting queries that define halfspaces 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, or have three points on a line as in S_3, or have multiple co-located points, none of which can be shattered by Q.

Lemma 4. *For any finite class Q, $VC - DIM(Q) \leq log|Q|$.*

Proof. If $VC - DIM(Q) = \alpha$, 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. □

This says that for any finite query class, the VC dimension is not too large. Returning to our small DB bounds, we can plug in $VC - DIM(Q)$ instead of $log|Q|$, and by this lemma, it can only be an improvement.

Theorem 5. *For any finite class of queries Q, if $R = \{y \in \mathbb{N}^{|\mathcal{X}|} | \|y\|_1 = O(\frac{VC-DIM(Q)}{\alpha^2})\}$, then $\forall x \in \mathbb{N}^{|\mathcal{X}|}, \exists y \in R \quad s.t. \quad \max_{f \in Q} |f(x) - f(y)| \leq \alpha$.*

This result is the analog of the theorem saying that there is a "good" "small" y, where now "small" depends on $VC - DIM(Q)$ instead of $log|Q|$. We can plug in this result to get overall SmallDB accuracy.

3 Private Multiplicative Weights

SmallDB required us to know all our queries in advance. We will now see a mechanism, Private Multiplicative Weights (MW), which allows queries to be chosen adaptively. This mechanism will be a combination of Sparse Vector (to answer threshold queries adaptively) and exponential gradient descent for learning linear predictors online. This latter algorithm is also known as Hedge or multiplicative weights. See [AHK12] for an excellent survey.

3.1 High-level Summary of Private MW

View database $x \in \mathbb{N}^{|\mathcal{X}|}$ as histogram, and consider only linear queries (i.e. linear functions of histogram). Then answering linear queries is the same problem as learning the linear function x that define query answers $< x, q >$, given a query $q \in [0, 1]^{|\mathcal{X}|}$. This is the same thing we did for SmallDB: we said, "I'm not going to give you answers to your queries. Instead I'll give you a database and you can answer your own queries. " Here this is handy because that problem can be phrased as learning a linear function, and we have lots of tools for solving that problem, such as multiplicative weights.

Multiplicative weights can learn any linear predictor by making only a small number of queries (good for our privacy budget). It maintains a "hypothesis predictor" and access the data only be requiring examples of queries where the hypothesis differs greatly from the true data. We can do this privately by Sparse Vector.

3.2 Regular MW

We will think of a database x as being a probability distribution over data universe $\mathcal{X}$. That is, let $\Delta([\mathcal{X}])$ denote the set of all distributions over the set $[|\mathcal{X}|]$, we have $x \in \Delta([\mathcal{X}])$. MW is instantiated with a learning rate parameter $\eta \leq 1$. (In later analysis, we will set $\eta = \alpha/2$.)

Algorithm 1 MW Update Rule

MW(x^+, f_t, v_t)

if $v_t < f_t(x^+)$ **then**

Let $r_t = f_t$

else

Let $r_t = 1 - f_t$

end if

Update: For all $i \in |\mathcal{X}|]$, let $\hat{x}_i^{t+1} = exp(-\eta r_t[i])x_i^t$, $x_i^{t+1} = \frac{\hat{x}_i^{t+1}}{\sum_{j=1}^{|\mathcal{X}|}} \hat{x}_j^{t+1}$

Output x^{t+1}

References

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(34):211–407, 2014.

[Rot10] Aaron Roth. Differential privacy and the fat-shattering dimension of linear queries. In *APPROX-RANDOM*, volume 6302, pages 683–695. Springer, 2010.