The algorithmic foundations of adaptive data analysis: outline

Kelvin Li

October 18, 2019

Abstract

This is the outline of the course *The algorithmic foundations of adaptive data analysis* given by Aaron Roth at the University of Pennsylvania.

The fundamental problem of machine learning is to study the underlying (unknown) distribution $\mathcal{D}$ through samples from $\mathcal{D}$. Learning is often an optimization problem that tries to minimize the generalization error[1] w.r.t. a loss function. *Overfitting* (or *false discovery*) is a problem that people try but not always avoid during learning (or experiment). The more overfitting occurs, the larger the generalization error. In machine learning, models are trained and tested using non-overlapping training and testing data sets respectively to avoid overfitting (sometimes even validation set to tune parameters). In experimental science, statistical hypotheses are stated up-front before collecting data (a.k.a., pre-registration) to avoid *p-hacking (p-fishing)*.

The way how research is done today is through an *adaptive* fashion, where the current trained model or hypothesis is selected based on some (if not all) of the results achieved in previous studies. In other words, the current model or hypothesis is a function of the data (indirectly). If the learning process is not treated properly, there is a high risk that the model will overfit or the discovery is a false finding.

Given a class $\mathcal{Q}$ of queries (or functions), using VC-dimension one is able to work out the sample complexity in order for the entire class of queries to generalize to the underlying distribution. In such a case, regardless a set of k queries are chosen adaptively or nonadptively from $\mathcal{Q}$ by an analyst, these queries will always generalize. The focus of adaptive data analysis is to prove generalization bound without making any assumptions about the query class $\mathcal{Q}$. This relies on the result that **stability implies generalization**.

This course studies some of the fundamental topics in *adaptive data analysis*, including some of the basic results from *statistical learning theory* and *differential privacy*. The entire course consists of 20 lectures.

[1] Generalization error is the difference between the expected and empirical errors.

1 Preliminary

1.1 Lecture 1

Throughout this course, we use a (SQL) *query* to replace the concept of learning or experiment. In particular, we focus on *statistical queries*. Let $\mathcal{X}$ denote an input domain, X a random variable, $\mathcal{D}$ denote a probability distribution of $\mathcal{X}$ and $S \sim \mathcal{D}$ denote a data set whose records are sampled *i.i.d.* from $\mathcal{D}$.

Definition 1.1. *A **statistical query** is a function $\phi : \mathcal{X} \to [0, 1]$ such that the value of ϕ on a distribution $\mathcal{D}$ is*

$$\phi(\mathcal{D}) = E_{X \sim \mathcal{D}}(\phi(X)) \tag{1}$$

and the value of ϕ on a data set S is

$$\phi(S) = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i). \tag{2}$$

The former is also known as *population parameter* and the latter is known as *sample parameter*.

The rest of this lecture introduces some of the basic concentration inequalities that provide bounds on how a random variable deviates from its expected value (or other values that we are interested in). The reason to bring in these inequalities is because we want to develop *mechanisms* that ensure a (adaptive or non-adaptive) query is accurate w.r.t. the data or underlying distribution with high probability.

Theorem 1.1. *(**Markov inequality**) For any non-negative random variable X and any $a > 0$,*

$$P(X \geq a) \leq \frac{E(x)}{a}.$$

This result is not so useful for our purpose except it derives the Chebyshev's inequality.

Theorem 1.2. *(**Chebyshev's inequality**) For any random variable X with mean $\mu = E[X]$ and variance $\sigma^2 = E[(X - \mu)^2]$, we have*

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}.$$

Here is an application of Chebyshev's inequality. Let $\sigma^2 = \frac{1}{4n}$ and $k^2 = \frac{1}{\beta}$, we get

$$P\left(|X - \mu| < \sqrt{\frac{1/\beta}{4n}}\right) \geq 1 - \beta.$$

To obtain a small error $|X - \mu| < \alpha$, it suffices to have $\sqrt{\frac{1/\beta}{4n}} \leq \alpha$. Solve for n, we get the *sample complexity* $n \geq \frac{1/\beta}{4\alpha^2}$. Both α and β are arbitrarily small. The sample size increases linearly with $1/\beta$. Here, α, β can be considered as error and the chance of making such an error.

A better bound is the *Chernoff bound* that uses the assumption that each trial is independent. There are several variations of Chernoff bound, based on slightly different assumptions.

Theorem 1.3. ***(Chernoff bound)*** *Let $X := \sum_{i=1}^{n} X_i$ where X_i are i.i.d . random variables taking values in $[0, 1]$ and $\mu = E[X]$. Then*

$$P(|X - \mu| \geq t) \leq 2e^{-2t^2/n}.$$

To obtain $P(|X - \mu| \leq \alpha) \geq 1 - \beta$, it suffices if the sample complexity $n \geq \frac{\ln 2/\beta}{2\alpha^2}$. This is an improvement over the previous result, since the sample size increases logarithmically with $1/\beta$.

1.2 Lecture 2

Define a query answering *mechanism* M that takes a query ϕ_i and a data set $S \sim \mathcal{D}$ as inputs and outputs an answer a_i. There are different types of mechanisms, such as the *empirical mechanism* that answers a query using the sample mean $a_i = \phi_i(S)$, the *noise addition mechanism* that adds some noise to the sample mean $a_i = \phi_i(S) + Lap(\cdot)$, etc.

The performance of a mechanism is measured by the worst absolute error w.r.t. the distribution $\mathcal{D}$ (or sample $S \sim \mathcal{D}$).

Definition 1.2. *A query answering mechanism M is (α, β)-**accurate** w.r.t. the underlying distribution $\mathcal{D}$ for k adaptively chosen statistical queries $K = \{q_1, \ldots, q_k\} \subseteq Q$ from a query class if for its answers $\{a_1, \ldots, a_k\}$ the following holds*

$$P\left(\sup_{q_i \in K} |q_i(\mathcal{D}) - a_i| \leq \alpha\right) \geq 1 - \beta.$$

The definition of (α, β)-**accurate** w.r.t. the data set $S \sim \mathcal{D}$ is defined analogously by replacing $q_i(\mathcal{D})$ with $q_i(S)$.

Theorem 1.4. ***(Union bound)*** *For every set of k events $E_1, \ldots, E_k$ in the same probability space, the probability of their union is at most the sum of their probabilities*

$$P\left(\bigcup_{j=1}^{k} E_j\right) \leq \sum_{j=1}^{k} P(E_j).$$

By Chernoff bound and Union bound, we can obtain the following bound for k nonadaptive statistical queries.

Theorem 1.5. *(**Nonadaptive queries**) Let $\mathcal{D}$ be a distribution on the set $\mathcal{X}$ and $\phi_1, \ldots, \phi_k$ be statistical queries with expectations $\mu_j = E_{\mathcal{D}}[\phi_j]$. If $S \sim D^n$ is a data set of size n drawn i.i.d. from $\mathcal{D}$, then*

$$P\left(\max_{j=1,\ldots,k} |E_S[\phi_j] - \mu_j| \leq \sqrt{\frac{\ln(2k/\beta)}{2n}}\right) \geq 1 - \beta.$$

The sample complexity is $n \geq \frac{\ln(2k/\beta)}{2\alpha^2}$, which increases logarithmically with the number of queries k, given α and β are fixed. For example, if $n = 100000$ and $\alpha = \beta = 0.01$, we can ask $k = 2425826$ nonadaptive queries without violating the probability bound.

1.3 Lecture 3

Adaptive queries can easily overfit. For example, if a data set $S = \{1, 2, 3, 4\}$, then construct the first query $q_1(S) = binary(\sum_{x \in S} 1/2^x)$ to output the binary representation of the sum of each $1/2^x$. This is a histogram of the data set S. Then construct the second query to overfit.

Remark 1.1. *Overfitting (or adversarial attack) appears when learning the identity of each record in a data set S, then carefully tailoring the subsequent queries to perfectly fit (or attack) S. So any method of answering statistical queries which promises protection from overfitting must in some sense prevent the analyst (or adversary) from learning too much about too many individual data points in S.*

To get a probability bound on the worst error of adaptive queries, we cannot just use the worst error among the k queries that have been asked, because the queries are not stated up-front like in the nonadaptive case. To use Chernoff bound and union bound, the union should be taken over all possible queries that could have been asked, had the answers of the previous queries come up differently. Assuming each of the k queries that was asked is a binary function. The space of all queries can be thought as a *Garden of Forking Paths* (or decision tree). The k queries that were asked is a particular path on the decision tree. But there are in total 2^k possible combinations of queries that could have been asked. Therefore, by the same argument as in Theorem 1.5 (i.e., Chernoff bound and Union bound), the sample complexity of adaptive queries is $n \geq \frac{k + \ln 2/\delta}{2\epsilon^2}$, which scales linearly with the number of queries k. The problem is that it is not always plausible to list all possible queries that could have been asked in order to find the worst error. So what can we do? In addition, Can we improve the sample complexity?

2 Description length bounds

Develop *transcript compressible* statistical estimators that allow us to control how much the analyst overfits the data, while allowing the analyst to still perform useful analyses.

Here, description length is the two part description length (according to **Occam's Razor**), where the first part corresponds to the model itself (including the number of parameters and their precision) and the second part corresponds to the compression of the data set given the model. The idea is that shorter description length estimators have lower generalization error. This can be thought from the model complexity perspective where an overly complicated model has longer description length and also higher risk of capturing random noise in the data. It can also be understood from the privacy perspective, where an overly complicated model fits every data point almost perfectly, hence reveals too much information about the data set, which then leads to a higher risk of being attacked.

2.1 Lecture 4

Let $\mathcal{A}, \mathcal{O}, \mathcal{Q}$ and S denote an analyst, a statistical estimator, a space of statistical queries and a data set respectively. For $i \in [1, k]$, the analyst $\mathcal{A}$ chooses a query $q_i \in \mathcal{Q}$ and gives it to the estimator $\mathcal{O}$. The estimator $\mathcal{O}$ generates an answer a_i to q_i based on the data set S and gives a_i to the analyst $\mathcal{A}$. Denote this **interaction** between $\mathcal{A}$ and $\mathcal{O}$ by $GT(\mathcal{A}, S, \mathcal{O}, \mathcal{Q})$.

Definition 2.1. *The output $T = (q_1, a_1, \ldots, q_k, a_k)$ of the interaction $GT(\mathcal{A}, S, \mathcal{O}, \mathcal{Q})$ is called the **transcript**.*

Definition 2.2. *A statistical estimator $\mathcal{O}$ is (α, β)-**accurate** w.r.t. the underlying distribution $\mathcal{D}$ for k queries from $\mathcal{Q}$ if for every algorithm $\mathcal{A}$ and for every distribution $\mathcal{D}$, the transcript T generated by $GT(\mathcal{A}, \mathcal{D}, \mathcal{O}, \mathcal{Q})$ satisfies*

$$P\left(\max_{i=1}^{k} |q_i(\mathcal{D}) - a_i| \leq \alpha\right) \geq 1 - \beta.$$

(α, β)-**accurate** w.r.t. the data set $S \sim \mathcal{D}$ can be defined similarly by replacing $q_i(\mathcal{D})$ by $q_i(S)$. Although we want to design random estimators, but it suffices to look at deterministic estimators and prove they are accurate.

Definition 2.3. *A statistical estimator $\mathcal{O}$ for k queries from $\mathcal{Q}$ is **transcript compressible** to $b(n, k)$ bits if for every analyst $\mathcal{A}$ there is a set of transcripts H_A of size $|H_A| \leq 2^{b(n,k)}$ such that for every data set S*

$$P(GT_{n,k}(\mathcal{A}, S, \mathcal{O}, \mathcal{Q}) \in H_A) = 1.$$

Theorem 2.1. *Any $b(n, k)$-compressible estimator $\mathcal{O}$ for statistical queries will have the property that for every data analyst $\mathcal{A}$ and every distribution $\mathcal{D}$, if $S \sim \mathcal{D}$ and $h = GT(\mathcal{A}, S, \mathcal{O}, \mathcal{Q})$, then the following is satisfied*

$$P\left(\max_{i=1}^{k} |q_i(S) - q_i(\mathcal{D})| \leq \sqrt{\frac{(b(n,k)+1)\ln 2 + \ln(k/\beta)}{2n}}\right) \geq 1 - \beta.$$

This theorem tells us that the queries (generated by the estimator) do not substantially overfit the data set S, but we do not know if the answers to these

queries are accurate w.r.t the distribution, because we do not compare a_i and $q_i(\mathcal{D})$. For example, we can easily construct a estimator that always return a constant value. This estimator has description length 0, but is not accurate at all.

The following transfer theorem is the main result for this section. A similar transfer theorem can also be proved for differential private mechanism (later in this course).

Theorem 2.2. *(**Transcript compressibility transfer theorem**) For any $\beta'' > 0$, a statistical estimator $\mathcal{O}$ for statistical queries that is*

1. *$b(n,k)$-compressible and*
2. *(α', β')-accurate w.r.t. a data set $S \sim \mathcal{D}$*

is (α, β)-accurate w.r.t. the distribution $\mathcal{D}$, where $\beta = \beta' + \beta''$ and

$$\alpha = \alpha' + \sqrt{\frac{(b(n,k)+1)\ln 2 + \ln(k/\beta'')}{2n}}$$

Note that the theorem is also applicable to *low sensitivity queries*.

2.2 Lecture 5

Two important characteristics that compressible estimators must have are the *robustness to post-processing* and *composability*.

Robustness to post-processing guarantees that an analyst, without additional knowledge about the private data set, cannot compute a function of the output of a statistical estimator and make its second part description length longer. That is, compress/fits the data set better.

Theorem 2.3. *(**Post-processing for transcript compressibility**) Suppose $\mathcal{O} : \mathcal{Q} \to \mathcal{R}$ is b-transcript compressible. Let $f : \mathcal{Q} \cup \mathcal{R} \to \mathcal{Q} \cup \mathcal{R}$ be an arbitrary stateful algorithm. Then $f \circ \mathcal{O}$ is also b-transcript compressible.*

Composability ensures that if all estimators are transcript compressible, so will their combination. Hence, we can build more sophisticated estimators using a combination of several compressible estimators. Later in the course we will see that given the privacy budget, we can work out the privacy budget for each query.

Theorem 2.4. *(**Composition for transcript compressibility**) Suppose $\mathcal{O}_1 : \mathcal{Q} \to \mathcal{R}$ is transcript compressible to $b_1(n, k_1)$ bits, and $\mathcal{O}_2 : \mathcal{Q} \to \mathcal{R}$ is transcript compressible to $b_2(n, k_2)$ bits. Then the composition $(\mathcal{O}_1, \mathcal{O}_2)$ is transcript compressible to $b(n, k_1 + k_2) = b_1(n, k_1) + b_2(n, k_2)$ bits.*

An example is the statistical estimator that returns the empirical mean of statistical queries, but to a truncated number of digits of precision. By truncating the digits, we reduced the first part of the description length (i.e., reduced

the model complexity). It can be proved that this estimator is accurate as well as lowering the risk of overfitting. For example, apply this estimator to the binary representation example before. We will not get the exact histogram of the records.

2.3 Lecture 6

More examples of accurate transcript compressible statistical estimators.

3 Stability and generalization (Lecture 7 - 10)

3.1 Algorithmic stability

In general, an algorithm is *stable* if changing one record in the input data will not change much of its output. Define stability in the context of classification.

Definition 3.1. *A deterministic algorithm M is* ***ϵ-uniform change-one (ϵ-UCO) stable*** *if for all data set S and S' that differ in one record, and for all inputs $z \in \mathcal{X}'$,*

$$|h_S(z) - h_{S'}(z)| \leq \epsilon,$$

where $h(\cdot)$ is a hypothesis/classifier and z is a point to be classified.

Generally speaking, a simpler classifier is more stable than a complicated one. Think about a linear and higher order classifiers. Denote the distributional and empirical accuracy respectively by

$$acc_{\mathcal{D}}(h) = 1 - E_{\mathcal{D}}[|\hat{y} - h_S(\hat{x})|]$$
$$acc_S(h) = 1 - \frac{1}{n}\sum_{i=1}^{n}(|y_i - h_S(x_i)|).$$

The next theorem proves that stability implies generalization.

Theorem 3.1. *Let M be an ϵ-UCO stable algorithm. For every data distribution $\mathcal{D}$ over labelled pairs in $\mathcal{X} \times \{0, 1\}$, the expected generalization error of the classifier satisfies*

$$|E_{S\sim\mathcal{D}}[acc_S(h_S) - acc_{\mathcal{D}}(h_S)]| \leq \epsilon$$

In other words, patterns that are recognized in the training data by a stable classifier are likely to appear in the underlying distribution too.

Adaptive data analysis concerns the interaction between an analyst and a query answering mechanism. Even the mechanism is guaranteed to be stable (e.g., a stable classification algorithm), can we guarantee that the analyst and mechanism pair is also stable after a certain number of interactions? **That is, is a stable mechanism closed under post-processing?**

3.2 Distributional stability

To answer the above question, we look at the distribution of the outputs of an mechanism and comparing this distribution with another output distribution from the same mechanism with a *neighbouring data set* (i.e., different by one record). Before doing so, we introduce three "distance measures" between distributions.

Let P and Q be two distributions on some set $\mathcal{Y}$ and share the same set of events for which probabilities are defined.

Definition 3.2. *The **total variation distance** between P and Q is*

$$\begin{aligned} d_{TV}(P,Q) &= \sup_{E\subseteq\mathcal{Y}} |P(E) - Q(E)| \\ &= \frac{1}{2}\int_{y\in\mathcal{Y}} |P(y) - Q(y)|dy. \end{aligned}$$

TVD is symmetric and always lies in $[0,1]$. It also satisfies the triangle inequality $d_{TV}(P,R) \leq d_{TV}(P,Q) + d_{TV}(Q,R)$.

A flips a fair coin and samples a point $x \sim P$ if the outcome is head and $x \sim Q$ otherwise. B tries to guess the outcome of the flip from the point x. The success probability of B's best strategy in the game is $\frac{1}{2}(1 + d_{TV}(P,Q))$. If P, Q are non-overlapping, the success probability is 1. If they are completely overlapping, the success probability is 0.

Definition 3.3. *The **multiplicative distance** between P and Q is*

$$d_\diamond(P,Q) = \sup_{E\subseteq\mathcal{Y}} \left|\ln \frac{P(E)}{Q(E)}\right| = \sup_{y\in\mathcal{Y}} \ln \frac{P(E)}{Q(E)}.$$

MD is symmetric, satisfies triangle inequality and lies in $[0,\infty)$. It upper bounds the TVD by $d_{TV}(P,Q) \leq \frac{1}{2}\left(e^{d_\diamond(P,Q)} - 1\right)$.

Definition 3.4. *The **KL-divergence** between P and Q is*

$$D_{KL}(P||Q) = \int_{y\in\mathcal{Y}} P(y) \ln \frac{P(y)}{Q(y)}.$$

KLD is **NOT** symmetric. It is nonnegative. Below are some relations between these three distance measures.

Lemma 3.1. *For any two distributions P and Q,*

1. $d_{TV}(P,Q) \leq \sqrt{\frac{1}{2} D_{KL}(P||Q)}$.
2. *If $d_\diamond(P,Q) \leq \epsilon$, then $D_{KL}(P||Q) \leq \epsilon(e^\epsilon - 1)$.*

The key result in this section is that all three measures remain the same stability after post-processing.

Lemma 3.2. *Consider a randomized algorithm A that maps elements in $\mathcal{Y}$ to (distributions over) elements in $\mathcal{Z}$. Then for any two random variables X, Y taking values in $\mathcal{Y}$, we have*

$$d_{TV}(A(X), A(Y)) \leq d_{TV}(X, Y)$$
$$d_{\diamond}(A(X), A(Y)) \leq d_{\diamond}(X, Y)$$
$$D_{KL}(A(X), A(Y)) \leq D_{KL}(X, Y).$$

For simplicity, use the $d_{TV}(X, Y)$ to denote the TVD between the distributions of X, Y.

Define stability in terms of the above measures.

Definition 3.5. *An randomized algorithm M is ϵ-TV stable if for all pairs of neighbouring data sets S and S', we have*

$$d_{TV}(M(S), M(S')) \leq \epsilon.$$

The other two are defined similarly. Note, stability w.r.t. multiplicative distance is the the same as ϵ-differential privacy as we will see later in the course.

3.3 Distributional stability implies generalization

Based on the results from the previous section, we can prove that if M is a stable algorithm, then the interaction between an algorithm A and M has low generalization error.

Theorem 3.2. *Let M be ϵ-TV stable and A be any algorithm that uses the output of M to decide on a statistical query $q_S = A(M(x))$. Then for any domain $\mathcal{X}$ and distribution $\mathcal{D}$*

$$|E_{S\sim\mathcal{D}}[q_S(S) - q_S(\mathcal{D})]| \leq \epsilon.$$

There is something that I do not understand in the lecture slides! It is about the expectation over the coins of M? Since the measures are non-increasing under post-processing, the theorem can be simplified by dropping the algorithm A.

Theorem 3.3. *Let M be a ϵ-TV stable algorithm which takes a data set $S \in \mathcal{X}^n$ as input and outputs a statistical query $q_S = M(x)$. Then for any domain $\mathcal{X}$ and distribution $\mathcal{D}$*

$$|E_{S\sim\mathcal{D}}[q_S(S) - q_S(\mathcal{D})]| \leq \epsilon.$$

The following theorem bounds the expectation of the absolute value of the error.

Theorem 3.4. *Let M be a ϵ-TV stable algorithm which takes a data set $S \in \mathcal{X}^n$ as input and outputs a statistical query $q_S = M(x)$. Then for any domain $\mathcal{X}$ and distribution $\mathcal{D}$*

$$E_{S\sim\mathcal{D}}[|q_S(S) - q_S(\mathcal{D})|] \leq \epsilon + \frac{2}{\sqrt{n}}.$$

Same as compressible estimators, we can also prove a transfer theorem for stable mechanisms. That is, if a mechanism is stable and accurate on samples, then it is also accurate on distribution.

Theorem 3.5. *(**Transfer theorem for TV-stable mechanisms**) If M is ϵ-TV stable and has expected worst case empirical error at most α, then for every distribution $\mathcal{D}$ and for every analyst A, when $S \sim \mathcal{D}$, the expected population error of the mechanism is*

$$E_{S\sim\mathcal{D}}\left[\max_{j=1}^{k} |a_j - q_j(\mathcal{D})|\right] \leq \epsilon + \alpha$$

Lemma 3.3. *If M is ϵ-TV stable, then for every distribution $\mathcal{D}$ and for every analyst A, when $S \sim \mathcal{D}$, the expected maximum generalization error of the mechanism is*

$$E_{S\sim\mathcal{D}}\left[\max_{j=1}^{k} |q_j(S) - q_j(\mathcal{D})|\right] \leq \epsilon + \sqrt{\frac{\log k}{n}}.$$

3.4 Composition of distributional notions

Suppose we have an interactive mechanism M that interacts with an analyst over k rounds. We can break it into k separate mechanisms $M_1, M_2, \ldots, M_k$, where each of the mechanisms takes as input the original data set S and the query q_i. We call M the *adaptive (sequential) composition* of $M_1, \ldots, M_k$.

Theorem 3.6. *(**Distributional stability notions compose adaptively**) Let $M = (M_1, \ldots, M_k)$ be the adaptive sequential composition of k mechanisms. If each M_i is ϵ-TV (or ϵ-KL or ϵ-MD) stable, then the adaptive composition M is $k\epsilon$-TV (or ϵ-KL or ϵ-MD) stable.*

The rest of these lectures are omitted.

4 Differential privacy and adaptive data analysis

4.1 Lecture 11

Definition 4.1. *A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is (ϵ, δ)-**differentially private** if for all pairs of data sets S, S' that differ in one record, for all events $E \subseteq \mathcal{Y}$,*

$$P(M(S) \in E) \leq e^{\epsilon} P(M(S') \in E) + \delta$$

Differentially private algorithms are distributional stable, and hence provide generalization guarantees.

Definition 4.2. *A query $q : \mathcal{X}^n \to \mathbb{R}^d$ is Δ-**sensitive** in the l_1 norm if for all pairs of neighbouring data sets $S = (x_1, \ldots, x_i, \ldots, x_n)$ and $S' = (x_1, \ldots, x_i', \ldots, x_n)$, the following is satisfied*

$$||q(S) - q(S')||_1 \leq \Delta$$

All statistical queries are Δ-sensitive with $\Delta = 1/n$.

Laplace mechanism Input: Data set $S = (x_1, \ldots, x_n) \in \mathcal{X}^n$ and parameter $\epsilon > 0$.

1. Receive a Δ-sensitive query $q : \mathcal{X} \to [0, 1]$ from analyst.
2. Output $a_i = \frac{1}{n} \sum_{i=1}^{n} q(x_i) + Lap(0, \frac{\Delta}{\epsilon})$.

Lemma 4.1. *The Laplace mechanism is ϵ-differentially private. If repeated k times, the Laplace mechanism is $k\epsilon$-differentially private.*

4.2 Lecture 12

Sparse vector technique (omit)

4.3 Lecture 13

The key result is that (ϵ, δ)-differential privacy satisfies a "strong composition" theorem, in which the ϵ parameter increases only with the square root of the number of queries. In other words, if we allow this much (i.e., ϵ) privacy to be revealed by each query, then the total privacy revealed after asking k queries is at most $O(\sqrt{k}\epsilon)$, which is much safer than $k\epsilon$.

Theorem 4.1. *(**Strong composition**) For all $\epsilon, \delta \geq 0$ and $\delta' > 0$, the adaptive composition of k algorithms, each of which is (ϵ, δ)-differentially private, is $(\hat{\epsilon}, \hat{\delta})$-differentially private where $\hat{\epsilon} = \epsilon\sqrt{2k \ln(1/\delta')} + k\epsilon\frac{e^{\epsilon}-1}{e^{\epsilon}+1}$ and $\hat{\delta} = k\delta + \delta'$.*

Another key feature of differentially private mechanisms is that they are **closed under post-processing**.

4.4 Lecture 14

The key result in this section is that for differentially private mechanisms, the expected generalization error can be converted to high probability bounds on low generalization error.

Theorem 4.2. *(**High probability bound**) Let $\epsilon \in [\sqrt{12/n}, 1/5]$ and $\delta \leq \epsilon/16$. Let $M : \mathcal{X}^n \to \mathcal{Q}_{1/n}$ be (ϵ, δ)-max-KL stable where $\mathcal{Q}_{1/n}$ is the class of $1/n$-sensitive queries $q : \mathcal{X}^n \to \mathbb{R}$. Let $\mathcal{D}$ be a distribution on $\mathcal{X}$. Then*

$$P\left(|q(\mathcal{D}) - q(S)| \geq 6\epsilon\right) = \max\left\{\frac{4\delta}{\epsilon}, e^{-\epsilon^2 n/8}\right\}.$$

A corollary of this theorem is the *transfer theorem* for differentially private mechanisms.

Corollary 4.1. *(**Transfer theorem for differentially private mechanisms**) For every distribution $\mathcal{D}$ and ϵ, δ as in the above theorem, if M is a $(\epsilon, \epsilon \cdot \delta)$-differentially private mechanism that answers adaptive statistical queries and it is (ϵ, δ)-accurate w.r.t. samples, then it is $(O(\epsilon), O(\delta))$-accurate w.r.t. the distribution $\mathcal{D}$.*