The Algorithmic Foundations of Adaptive Data Analysis October 13, 2017

Lecture 7–10: Stability and Adaptive Analysis I

Lecturer: Adam Smith *Scribe: Adam Smith*

So far, we've seen that mechanisms whose output is compressible do not allow overfitting: if the mechanism's output is compressible to b bits then, for any given (deterministic) analyst, there are at most 2^b sets of queries that can actually arise, and we can take a union bound over all $k \cdot 2^b$ queries that the analyst could ever make to get a bound on how much those queries' empirical means can deviate from their population means.

We'll see at least three other general approaches to limiting the bias over selected analyses: stability-based techniques, information bounds (which encompass compression and, to some extent, stability) and explicit conditioning. We'll start with stability, which provides a different formalism for limiting how the data can affect the output of an algorithm.

1 Algorithmic Stability

We'll start with the case of classification, and an application that is not directly about adaptivity. Suppose we have an algorithm M that takes a data set $\mathbf{s} = ((x_1, y_1), ..., (x_n, y_n))$ of labeled pairs in the domain $\mathcal{X} = \mathcal{X}' \times \{0,1\}$ and outputs a *hypothesis* $h : \mathcal{X}' \to [0,1]$ that associates, to every possible future point x, a probability that it should be labeled "1".

Definition 1 *A deterministic algorithm M is ϵ-uniform change-one (ϵ-UCO) stable if for all data set $\mathbf{s}$ and $\mathbf{s}'$ that differ in one element, and for all inputs $z \in \mathcal{X}'$,*

$$|h_{\mathbf{s}}(z) - h_{\mathbf{s}'}(z)| \leq \epsilon \qquad \text{where } h_{\mathbf{s}} = M(\mathbf{s}) \text{ and } h_{\mathbf{s}'} = M(\mathbf{s}')\,.$$

A classic example of a stable classification algorithm is the *k-nearest neighbors classifier* (t-NN). Here the example domain $\mathcal{X}'$ is $\mathbb{R}^d$ for some finite d. Given a data set and a new point, we classify the point using the average of its t nearest neighbors' labels:

```
Algorithm 1: $t$-NN$(\mathbf{s}, z)$
  Input: $\mathbf{s} = \{(x_i, y_i)\}_{i=1,...,n}$ is a collection of pairs in $\mathbb{R}^d \times \{0,1\}$;
  $z \in \mathbb{R}^d$ is a point to be classified.
  Let $i_1, ..., i_t$ be the indices of the $t$ points in $\mathbf{s}$ that are nearest to $z$
    (that is, that minimize $\|z - x_i\|$, breaking ties arbitrarily);
  return $h_{\mathbf{s}}(z) = \frac{1}{k}\sum_{j=1}^{t} y_{i_j}$
```

Proposition 2 *t-NN classification is $\frac{1}{t}$-UCO stable.*

Proof For every fixed point x and data set $\mathbf{s}$, changing a point in $\mathbf{s}$ changes at most one of the t nearest neighbors, so the average label can go up or down by at most $1/t$. ■

Recall that we would ideally like to the accuracy of a classifier with respect to fresh samples from the underlying distribution. For a classifier that makes "soft" predictions (i.e. outputs a probability in $[0,1]$), a simple measure is the expected absolute error:

$$\mathrm{acc}_{\mathcal{D}}(h) \overset{\mathrm{def}}{=} 1 - \mathop{\mathbb{E}}_{(\tilde{x},\tilde{y})\sim\mathcal{D}} (|\tilde{y} - h_{\mathbf{s}}(\tilde{x})|)\,.$$

As in Lecture 3, we'll use $\mathrm{acc}_{\mathbf{s}}(h)$ to denote the classifier's empirical accuracy.

$$\mathrm{acc}_{\mathbf{s}}(h) = 1 - \frac{1}{n}\sum_{i=1}^{n}(|y_i - h_{\mathbf{s}}(x_i)|)\,.$$

Theorem 3 *Let M be ϵ-uniform leave-one-out hypothesis stable. For every data distribution $\mathcal{D}$ over labeled pairs in $\mathcal{X} \times \{0,1\}$, the expected generalization error of the classifier is at most ϵ, that is:*

$$\left| \mathop{\mathbb{E}}_{\mathbf{s}\sim\mathcal{D}^n} \Big(acc_{\mathbf{s}}(h_{\mathbf{s}}) - acc_{\mathcal{D}}(h_{\mathbf{s}}) \Big) \right| \leq \epsilon$$

Why is this useful? The algorithm can, in a sense, check its own work: If the NN classifier does well on the data it was handed, then (for sufficiently large t) it will also do well on future unseen examples from the same distribution.

Remark For readers familiar with uniform convergence and VC dimension: A statement like Theorem 3 would follow directly from standard tools if the family of classifiers produced by NN were bounded in VC dimension. However, it is not, and there is no reason to think that it would satisfy uniform convergence over the entire family of classifiers. Rather, the theorem shows that *the classiifer that is actually output* will do approximately as well on fresh samples as it does on the data .

For the proof we introduce additional notation that will be helpful below. Given a data set $\mathbf{s}$, and a position i, let $\mathbf{s}_{-i}$ denote the data set from which $\mathbf{s}$'s ith entry has been removed and let $\mathbf{s}_{i\to x'}$ denote the data set in which the ith entry of $\mathbf{s}$ has been replaced by a new value x'.

Proof The theorem asks us to bound an absolute value; we'll prove only the upper bound, since the lower bound is symmetric.

$$\mathop{\mathbb{E}}_{\mathbf{s}\sim\mathcal{D}^n} \Big(\mathrm{acc}_{\mathbf{s}}(h_{\mathbf{s}}) - \mathrm{acc}_{\mathcal{D}}(h_{\mathbf{s}}) \Big) = \frac{1}{n} \sum_{i=1}^{n} \left(\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} |y_i - h_{\mathbf{s}}(x_i)| - \mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} |\tilde{y} - h_{\mathbf{s}}(\tilde{x})| \big) \right) \tag{1}$$

Here is where we will pull the big switch. In the expressions above, the joint distribution on $\mathbf{s}, (\tilde{x}, \tilde{y})$ consists of a sample of $n + 1$ points drawn i.i.d from $\mathcal{D}$. So in the second expectation, we can swap (x_i, y_i) and $(\tilde{x}, \tilde{y})$ without changing the distribution of the random variable.

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} |\tilde{y} - h_{\mathbf{s}}(\tilde{x})| \big) = \mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} |y_i - h_{\mathbf{s}_{i\to(\tilde{x},\tilde{y})}}(x_i)| \big) \qquad \leftarrow \text{(“The switch.” Watch the prover's hands carefully.)} \tag{2}$$

Substituting this back into the Equation (1), and combining the two expectations, we get

$$\begin{aligned} \mathop{\mathbb{E}}_{\mathbf{s}\sim\mathcal{D}^n} \Big(\mathrm{acc}_{\mathbf{s}}(h_{\mathbf{s}}) - \mathrm{acc}_{\mathcal{D}}(h_{\mathbf{s}}) \Big) &= \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} \big(|y_i - h_{\mathbf{s}}(x_i)| - |y_i - h_{\mathbf{s}_{i\to(\tilde{x},\tilde{y})}}(x_i)| \big) \\ &\leq \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} \big(|h_{\mathbf{s}}(x_i) - h_{\mathbf{s}_{i\to(\tilde{x},\tilde{y})}}(x_i)| \big) \end{aligned} \tag{3}$$

We have passed from evaluating the same classifier $h_{\mathbf{s}}$ on two different data sources (either $\mathbf{s}$ or fresh samples) **to evaluating different classifiers** (generated from either $\mathbf{s}$ or $\mathbf{s}_{i\to(\tilde{x},\tilde{y})}$) **on the same data point** (x_i, y_i). We can now invoke ϵ-UCO stability: changing one point in $\mathbf{s}$ changes the classifier's prediction by at most ϵ. Hence

$$\mathop{\mathbb{E}}_{\mathbf{s}\sim\mathcal{D}^n} \Big(\mathrm{acc}_{\mathbf{s}}(h_{\mathbf{s}}) - \mathrm{acc}_{\mathcal{D}}(h_{\mathbf{s}}) \Big) \leq \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ (\tilde{x},\tilde{y})\sim\mathcal{D}}} \Big(\epsilon \Big) = \epsilon \,.$$

Similarly, we can prove the symmetric lower bound: $\mathbb{E}_{\mathbf{s}\sim\mathcal{D}^n} \Big(\mathrm{acc}_{\mathbf{s}}(h_{\mathbf{s}}) - \mathrm{acc}_{\mathcal{D}}(h_{\mathbf{s}}) \Big) \geq -\epsilon$. ■

2 Distributional Stability

Let's return to the adaptive setting. There are now two algorithms involved: the mechanism and the analyst. Together they generate queries based on the data. We'd love to have guarantees along the lines of Theorem 3, but for that we would need the mechanism-analyst pair to be

Can we ever guarantee that the algorithm and mechanism together satisfy stability? Even if the mechanism itself is stable (say, for example, it releases a nearest neighbor classifier), the effect of the analyst may destroy that stability (for example, it could read the description of $\mathbf{s}$ embedded in the NN classifier and output an arbitrary function of the data). The key is to find a notion of stability that satisfies *postprocessing*, as did our notion of compressibility (c.f. Lecture 5). We want that if M is stable, then so is $A \circ M$, regardless of how A works. We can do this if we switch to randomized mechanisms, and consider the stability of the *distribution* on outputs of the mechanism.

2.1 Comparing probability distributions

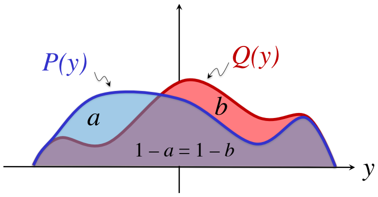
To do this, we'll need a way to measure how far apart two probability distributions are. We will in fact use multiple measures in this course. Suppose we have distributions P and Q on some set $\mathcal{Y}$. We assume that P and Q share the same "σ-algebra", or set of events for which probabilities are defined.

2.1.1 Total Variation Distance

The *total variation distance* (also called *statistical difference* or half of the L_1 *distance*) is $d_{TV}(P,Q) \overset{\text{def}}{=} \sup_{E\subseteq\mathcal{Y}} |P(E) - Q(E))|$. That is, When P and Q are discrete or have well-defined densities, we have

$$\begin{aligned} d_{TV}(P,Q) &\overset{\text{def}}{=} \sup_{E\subseteq\mathcal{Y}} |P(E) - Q(E))| \\ &= \underbrace{\int_{y:P(y)>Q(y)} |P(y) - Q(y)| dy}_{\text{Area } a \text{ in figure}} = \underbrace{\int_{y:P(y)<Q(y)} |P(y) - Q(y)| dy}_{\text{Area } b \text{ in figure}} = \underbrace{\frac{1}{2}\int_{y\in\mathcal{Y}} |P(y) - Q(y)| dy}_{a+b} . \end{aligned} \tag{4}$$

To see why the first equality holds, notice that the event E that maximizes $P(E)-Q(E)$ includes exactly the set of points y for which $P(y) > Q(y)$ (every such point helps, and other points either hurt or don't help). To see the second equality, note that areas a and b in the following figure are identical, since the area below both curves is equal to both $1-a$ and $1-b$..



The total variation distance is symmetric and always lies in $[0,1]$. It also satisfies the triangle inequality: for all distributions P,Q,R, we have $d_{TV}(P,R) \le d_{TV}(P,Q) + d_{TV}(Q,R)$.

Exercise 1 *Let $U_{[a,b]}$ denote the uniform distribution on the interval $[a,b]$. Fix $\epsilon > 0$. How close are $U_{[0,1]}$ and $U_{[\epsilon,1+\epsilon]}$? [Answer: $d_{TV}(U_{[0,1]}, U_{[\epsilon,1+\epsilon]}) = \epsilon$.]*

The total variation ditance has an important operational interpretation. Namle,y imagine a game between two players, Alice and Bob. Alice flips a coin out of Bob's sight and gets $C \in \{Heads, Tails\}$. If C is heads, Alice samples a point Z according to P. If C is tails, she samples Z according to Q. Now she shows Z to Bob, and Bob tries to guess C.

Lemma 4 *The success probability of Bob's best strategy in this game is $\frac{1}{2}(1 + d_{TV}(P,Q))$.*

Proof The probability that Bob sees a particular sample z is $\frac{P(z)+Q(z)}{2}$. Conditioned on seeing z, the probability that $C = Heads$ is therefore $\frac{P(z)}{P(z)+Q(z)}$. Bob's best strategy is therefore to guess "heads" whenever $P(z) > Q(z)$, and tails when $P(z) < Q(z)$ (his guesses when $P(z) = Q(z)$ don't change anything).

Conditioned on C being heads, Bob's probability of being corrrect is $P(E)$, where $E = \{y \in \mathcal{Y} : P(y) > Q(y)\}$. Similarly, his probability of being correct when $C = Tails$ is $Q(E^c)$. His overall probability of winning is therefore $\frac{1}{2}(P(E) + Q(E^c)) = \frac{1}{2}(1 + P(E) - Q(E)) = \frac{1}{2}(1 + d_{TV}(P, Q))$. ■

2.1.2 Multiplicative Distance (or "differential privacy metric")

The *multiplicative distance* $d_\diamond(P, Q) \overset{\text{def}}{=} \sup_{E \subseteq \mathcal{Y}} \left| \ln\left(\frac{P(E)}{Q(E)}\right) \right| = \sup_{y \in \mathcal{Y}} \ln\left(\frac{P(y)}{Q(y)}\right)$.

This is a much more strict version of the total variation distance: distributions that are very different on even a tiny fraction of their domain will be far apart in this metric.

The multiplicative distance upper bounds the total variation distance. since for every event E, we have $P(E) \le e^{d_{TV}(P,Q)} Q(E)$, and so $P(E) - Q(E) \le Q(E) \cdot (e^{d_{TV}(P,Q)} - 1) \le e^{d_{TV}(P,Q)} - 1$. (We can prove a slightly tighter bound by observing that the smallest of $P(E), P(E^c)$, $Q(E)$, $Q(E^c)$ is at most $\frac{1}{2}$. By making the smallest of these play the role of $Q(E)$ in the proof above, we get

$$d_{TV}(P, Q) \le \frac{1}{2}\left(e^{d_{TV}(P,Q)} - 1\right).$$

The multiplicative distance is always nonnegative, but it can be infinite (for example, if there exists a point where P has probability 0 but Q has nonzero probability).

Exercise 2

1. *What is $d_\diamond(U_{[0,1]}, U_{[\epsilon,1+\epsilon]})$? [Answer: ∞]*
2. *What is $d_\diamond(N(0, 1), N(\epsilon, 1))$? [Answer: ∞]*
3. *The Laplace distribution $Lap(\mu, \lambda)$ is given by the density function $f(x) = \frac{1}{2\lambda}\exp(-\frac{|x-\mu|}{\lambda})$. It has expectation μ and standard deviation $\sqrt{2} \cdot \lambda$. What is $d_\diamond(Lap(0, 1), Lap(\epsilon, 1))$? [Answer: ϵ.]*
   What does the previous answer imply about $d_{TV}(Lap(0, 1), Lap(\epsilon, 1))$? [Answer: at most $\frac{1}{2}e^\epsilon - 1$.]

As with d_{TV}, the mutliplicative distance is symmetric, nonnegative and satisifes the triangle inequality.

2.1.3 Kullback-Liebler Divergence

The *KL divergence* between P and Q: $D_{KL}(P\|Q) \overset{\text{def}}{=} \int_{y \in \mathcal{Y}} P(y) \ln\left(\frac{P(y)}{Q(y)}\right) dy = \mathbb{E}_{Y \sim P}\left(\ln\left(\frac{P(y)}{Q(y)}\right)\right)$.

This measure is a bit trickier to work with, though we'll see it is very useful. First, it is not symmetric in P and Q! Second, the quantity $\ln\left(\frac{P(y)}{Q(y)}\right)$ can be negative, so it is not clear a priori that the KL divergence is always nonnegative. But it is!

Lemma 5 *For all distributions P, Q, $D_{KL}(P\|Q) \ge 0$, with equality if and only if $P = Q$.*

We defer that proof to later.

Exercise 3

1. *What is $D_{KL}(U_{[0,1]} \| U_{[\epsilon,1+\epsilon]})$? [Answer: $+\infty$.]*
2. *What is $D_{KL}(N(0, 1)\|N(\epsilon, 1))$? [Answer: $\frac{1}{2}\epsilon^2$. This follows from the following more general fact: $D_{KL}(N(0, \sigma^2)\|N(\mu, \sigma^2)) = \frac{1}{2}\left(\frac{\mu}{\sigma}\right)^2$. To see why, compute $\mathbb{E}(\ln(P/Q)) = \mathbb{E}_{X \sim N(0,\sigma^2)}(\frac{-X^2+(X-\mu)^2}{2\sigma^2}) = \mathbb{E}_{X \sim N(0,\sigma^2)}(\frac{\mu^2 - 2\mu X}{2\sigma^2}) = \frac{\mu^2}{2\sigma^2}$.]*
3. *What is $D_{KL}(Lap(0, 1)\|Lap(\epsilon, 1))$?*

Finally, we can relate D_{KL} to the other metrics via two nontrivial results:

Lemma 6 *For any two distributions P, Q:*

1. $d_{TV}(P,Q) \le \sqrt{\frac{1}{2} D_{KL}(P\|Q)}$ *("Pinsker's inequality").*
2. *If $d_\diamond(P,Q) \le \epsilon$, then $D_{KL}(P\|Q) \le \epsilon(e^\epsilon - 1)$.*

We defer this proof, too, to later.

2.2 Postprocessing and Stability

All three of these measures satisfy some form of monotonicity under postprocessing (often called "data processing" inequalities).

Lemma 7 *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*

$$d_{TV}(A(X), A(Y)) \le d_{TV}(X,Y) \tag{5}$$

$$d_\diamond(A(X), A(Y)) \le d_\diamond(X,Y) \tag{6}$$

$$D_{KL}(A(X), A(Y)) \le D_{KL}(X,Y)\,. \tag{7}$$

In the statements above, we commit an abuse of notation common in the research literature: Given random variables X and Y defined on the same set $\mathcal{Y}$, we will denote by $d_{TV}(X,Y)$ the total variation distance between the distributions of X and Y. Similarly for $d_\diamond$ and D_{KL}.

Proof To prove the lemma, first note that we can write $A(y,r)$ as a deterministic function which takes its main input (say y) and an additional source of randomness r whose distribution is independent of y.

Next, note that for each of these measures, the distance between X and Y is the same as the distance between the pairs (X,R) and (Y,R) where R is the randomness of A, which is indepdendent of X and Y.

(That is because each of the three measures above can be written in terms of distribution of the odds ratio $\frac{P_X(y)}{P_Y(y)}$. When we add in the extra independent random variable R, the probabilities of seeing a pair (y,r) become $P_X(y)P_R(r)$ and $P_Y(y)P_R(r)$, and so the odds ratio remains the same.)

We can prove the lemma for TV distance: Let E be any event in (think "subset of") $\mathcal{Z}$, and let $F = A^{-1}(E)$ denote the set of pairs $\{(y,r) : A(y,r) \in E\}$. Then

$$\begin{aligned} &Pr(A(X) \in E) - \Pr(A(Y) \in E) \\ &\qquad = \Pr\Big((X,R) \in F\Big) - \Pr\Big((Y,R) \in F\Big) \le d_{TV}\Big((X,R),\ (Y,R)\Big) = d_{TV}(X,Y)\,. \end{aligned}$$

Similarly, for $d_\diamond$, we have (for events E with nonzero probability under $A(Y)$),

$$\begin{aligned} &Pr(A(X) \in E)/\Pr(A(Y) \in E) \\ &\qquad = \Pr\Big((X,R) \in F\Big)/\Pr\Big((Y,R) \in F\Big) \le \exp\Big(d_\diamond\Big((X,R),\ (Y,R)\Big)\Big) = \exp(d_\diamond(X,Y))\,. \end{aligned}$$

Proving this for KL distance is a bit more delicate, and we again defer the proof. ■

Not all notions of distance on probability distributions are nonincreasing under postprocessing! For example, the L_2 distance $d_2(P,Q) \overset{\text{def}}{=} \int_y (P(y) - Q(y))^2 dy$ is popular in nonparametric statistics and signal processing, but is not even preserved by rescaling. (Exercise: Compare $d_2(U_{[0,1]}, U_{[0,2]})$ to $d_2(U_{[0,1/2]}, U_{[0,1]})$.)

Definition 8 *An randomized algorithm M is ϵ-TV* stable *if for all neighboring pairs of data sets* $\mathbf{s}$ *and* $\mathbf{s}'$*, we have*

$$d_{TV}(M(\mathbf{s}), M(\mathbf{s}')) \le \epsilon.$$

Similarly, we can define stability with respect to KL and $d_\diamond$.

```
Algorithm 2: Laplace mechanism($\epsilon, \mathbf{s}$)
  Input: Data set $\mathbf{s} = (x_1, ..., x_n) \in \mathcal{X}^n$ and parameter $\epsilon > 0$.
  Receive a statistical query $q : \mathcal{X} \to [0, 1]$ from analyst ;
  return $\frac{1}{n} \sum_{i=1}^{n} q(x_i) + Z$ where $Z \sim \mathrm{Lap}(0, \frac{1}{n\epsilon})$.
```

Stability with respect to $d_\diamond$ is also called *ϵ-differential privacy.* It has been extensively studied in its own right, and in future lectures we will return to it.

Lemma 9 *If used to answer a single query, the Laplace mechanism with parameter ϵ is ϵ-differentially private (same as ϵ-$d_\diamond$ stable). It is also $\frac{1}{2}(e^\epsilon - 1)$-TV stable.*

Proof When we change the input from $\mathbf{s}$ to a neighboring data set $\mathbf{s}'$ (that differs in one input), the emprical answer $q(\mathbf{s})$ changes by at most $\frac{1}{n}$. So we are comparing two Laplace distributions with scale parameter $1/(n\epsilon)$ and means that differ by $1/n$. The ratio of densities at any point z is therefore $\exp(-\epsilon n(|q(\mathbf{s}) - z| + |q(\mathbf{s}') - z|)) \leq \exp(\epsilon n |q(\mathbf{s}) - q(\mathbf{s}')|) \leq \exp(\epsilon)$. Therefore $d_\diamond(M(\mathbf{s}), M(\mathbf{s}'))$ is at most ϵ. ■

```
Algorithm 3: Gaussian mechanism($\sigma^2, \mathbf{s}$)
  Input: Data set $\mathbf{s} = (x_1, ..., x_n) \in \mathcal{X}^n$ and parameter $\epsilon > 0$.
  Receive a statistical query $q : \mathcal{X} \to [0, 1]$ from analyst ;
  return $\frac{1}{n} \sum_{i=1}^{n} q(x_i) + Z$ where $Z \sim N(0, \sigma^2)$.
```

Lemma 10 *If used to answer a single query, the Gaussian mechanism with parameter σ^2 is $\frac{1}{2(n\sigma)^2}$-KL-stable, and $\frac{1}{2n\sigma}$-TV-stable.*

Proof This follows from a similar reasoning to the Laplace lemma above: we are comparing $N(q(\mathbf{s}), \sigma^2)$ with $N(q(\mathbf{s}'), \sigma^2)$. Their KL divergence is at most $\frac{1}{2} \cdot \left(\frac{q(\mathbf{s}) - q(\mathbf{s}')}{\sigma}\right)^2 \leq \frac{1}{2(n\sigma)^2}$. ■

3 Distributional Stability and Generalization

Why are these notions of stability useful? Stable algorithms cannot overfit.

Theorem 11 *Let M be ϵ-TV stable and A be any algorithm that uses the output of M to decide on a statistical query $q_\mathbf{s} = A(M(x))$. Then for every domain $\mathcal{X}$ and distribution $\mathcal{D}$:*

$$\left| \underset{\substack{\mathbf{s} \sim \mathcal{D}^n \\ \textit{coins of } M, A}}{\mathbb{E}} \Big(q_\mathbf{s}(\mathbf{s}) - q_\mathbf{s}(\mathcal{D}) \Big) \right| \leq \epsilon \,.$$

Before proving this theorem, first note that we can remove A from the statement above. *Because d_{TV} can only decrease under postprocessing*, the composed mechanism $A \circ M$ is still ϵ-TV stable. The theorem above is therefore a corollary of the following simplified statement.

Theorem 12 (Simplified version of Theorem 11) *Let M be a ϵ-TV stable algorithm which takes a database $\mathbf{s} \in \mathcal{X}^n$ as input and outputs a statistical query $q_\mathbf{s} = M(x)$. Then for every domain $\mathcal{X}$ and distribution $\mathcal{D}$:*

$$\left| \underset{\substack{\mathbf{s} \sim \mathcal{D}^n \\ \textit{coins of } M}}{\mathbb{E}} \Big(q_\mathbf{s}(\mathbf{s}) - q_\mathbf{s}(\mathcal{D}) \Big) \right| \leq \epsilon \,.$$

We will need the following lemma during the proof:

Lemma 13 *For all X, Y on $\mathcal{Y}$ with $d_{TV}(X,Y) \le \epsilon$, and for all functions $f : \mathcal{Y} \to [0,1]$,*

$$|\mathbb{E}(f(X)) - \mathbb{E}(f(Y))| \le \epsilon\,.$$

Proof Let P_X, P_Y be the distributions of X, Y respectively.

$$\begin{aligned} E(f(X)) - E(f(Y)) = \int f(y)P_X(y)dy - \int f(y)P_Y(y)dy = \int f(y)(P_X(y) - P_Y(y))dy \\ \le \int_{y:P_X(y)>P_Y(y)} |P_X(y) - P_Y(y)|dy = d_{TV}(X,Y)\,. \quad (8) \end{aligned}$$

■

Proof (of Theorem 12): This proof is almost the same as the proof of Theorem 3. There are a few key differences. First, the data points are now abstract values in a domain $\mathcal{X}$; they need not be example-label pairs.

Second, the expectation must now be taken over the coins of M in addition to the choice of $\mathbf{s}$ and the new sample $\tilde{x}$.

The "switch" (Equation (2)) now looks like this:

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ \tilde{x}\sim\mathcal{D} \\ \text{coins of } M}} \big(q_{\mathbf{s}}(\tilde{x})\big) = \mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ \tilde{x}\sim\mathcal{D} \\ \text{coins of } M}} \big(q_{\mathbf{s}_{i\to\tilde{x}}}(x_i)\big) \tag{9}$$

Finally, to complete the proof we apply TV stability instead of UCO stability. Analogously to Equation (3), we must bound

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ \tilde{x}\sim\mathcal{D} \\ \text{coins of } M}} \Big(q_{\mathbf{s}}(x_i) - q_{\mathbf{s}_{i\to\tilde{x}}}(x_i)\Big)$$

which we can we rewrite to separate the coins of M:

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ \tilde{x}\sim\mathcal{D}}} \left(\mathop{\mathbb{E}}_{\text{coins of } M} q_{\mathbf{s}}(x_i) - \mathop{\mathbb{E}}_{\text{coins of } M} q_{\mathbf{s}_{i\to\tilde{x}}}(x_i)\right)\,.$$

For fixed x_i, the function $f_{x_i}(q) = q(x_i)$ is a bounded function (that takes a function as input and returns a value in $[0,1]$). Applying Lemma 13 (on how expectations change with changes in d_{TV}) and the TV-stability of M, we get that for every $\mathbf{s}, \tilde{x} \in \mathcal{X}^{n+1}$, we have

$$\mathop{\mathbb{E}}_{\text{coins of } M} q_{\mathbf{s}}(x_i) - \mathop{\mathbb{E}}_{\text{coins of } M} q_{\mathbf{s}_{i\to\tilde{x}}}(x_i) \le \epsilon\,.$$

The remainder of the proof is identical to that of Theorem 3. ■

This theorem bounds the expected generalization error (difference between empirical mean and population mean). But it is more natural to get bounds on the expectation of the absolute value of the error, $\Big|q_{\mathbf{s}}(\mathbf{s}) - q_{\mathbf{s}}(\mathcal{D})\Big|$. With some additional work, one can get the following bound, which we state without proof:

Theorem 14 *Let M be a ϵ-TV stable algorithm which takes a database $\mathbf{s} \in \mathcal{X}^n$ as input and outputs a statistical query $q_{\mathbf{s}} = M(x)$. Then for every domain $\mathcal{X}$ and distribution $\mathcal{D}$:*

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ \textit{coins of } M}} \Big|q_{\mathbf{s}}(\mathbf{s}) - q_{\mathbf{s}}(\mathcal{D})\Big| \le \epsilon + \tfrac{2}{\sqrt{n}}\,.$$

Exercise 4

1. *Show that stability on average over samples in $\mathcal{D}$ is sufficient for Theorem 12 : for $\mathbf{s}$ sampled i.i.d. from $\mathcal{D}$ and $\mathbf{s}'$ obtained by replacing a particular position with a fresh sample from $\mathcal{D}$, it should hold that the expected TV distance of $M(\mathbf{s})$ and $M(\mathbf{s}')$ is at most ϵ.*
2. *Show that Theorem 12 holds for mechanisms that output arbitrary low-sensitivity queries, not only linear queries.*

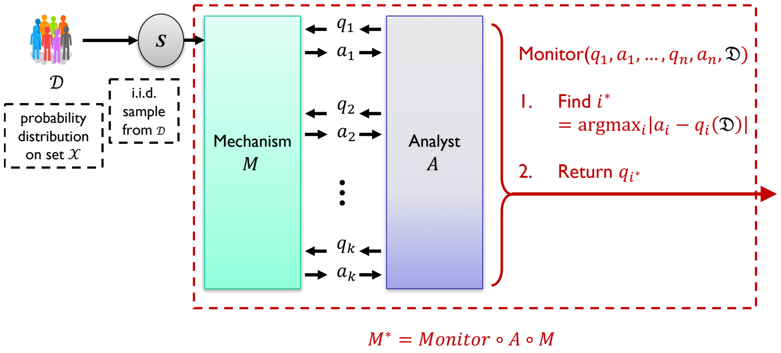

$\mathcal{D}$
probability distribution on set $\mathcal{X}$
i.i.d. sample from $\mathcal{D}$
s
Mechanism M
q_1
a_1
q_2
a_2
q_k
a_k
Analyst A
Monitor$(q_1, a_1, \ldots, q_n, a_n, \mathfrak{D})$
1. Find i^* $= \text{argmax}_i |a_i - q_i(\mathfrak{D})|$
2. Return q_{i^*}
$M^* = Monitor \circ A \circ M$

Figure 1: The "monitor argument" from the proof of Theorem 15.

3.1 Lifting to Many Rounds via the Monitor Argument

As with compression-based mechanisms, we can combine generalization guarantees with low empirical error to get overall error guarantees. The exact argument is different, however. In particular, the reduction from many rounds to a single round has a different flavor.

Theorem 15 (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 $\mathbf{s} \sim \mathcal{D}^n$, the expected population error of the mechanism is*

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ coins\ of\ M,A}} \left(\max_{j=1}^{k} |a_j - q_j(\mathcal{D})| \right) \le \epsilon + \alpha \,.$$

Proof We can write bound the total error by the sum of two terms:

$$\max_{j=1}^{k} |a_j - q_j(\mathcal{D})| \le \underbrace{\max_{j=1}^{k} |a_j - q_j(\mathbf{s})|}_{\text{empirical error}} + \underbrace{\max_{j=1}^{k} |q_j(\mathbf{s}) - q_j(\mathcal{D})|}_{\text{generalization error}} \,.$$

It is tempting to attempt to apply our generalization result directly in order to bound the second term. This turns out to be delicate (we will return to it!). Instead, we proceed via a different route.

The idea is to "lift" the bound we have for a single adaptively chosen query to the worst of a set of queries by considering a thought experiment. We imagine a new algorithm, called the *monitor*, which takes as input the final conversation $q_1, a_1, ..., q_k, a_k$ between M and A, as well as the distribution $\mathcal{D}$. The monitor, illustrated in Figure 1, outputs the name of the query with the worst generalization error:

$$Monitor(q_1, a_1, ..., q_k, a_k, \mathcal{D}) \overset{\text{def}}{=} \arg\max_{j} |a_j - q_j(\mathcal{D})| \,.$$

The monitor can only ever be a thought experiment, since it requires knowing the true distribution $\mathcal{D}$ in order to pick the worst query.

Let $M^* = Monitor \circ A \circ M$ be the equally fictional algorithm that takes as input $\mathbf{s}$ and the distribution $\mathcal{D}$, runs the interaction between $M(\mathbf{s})$ and A, and runs the monitor (to which it hands $\mathcal{D}$) on the result. *Because of closure under postprocessing, M^* is ϵ-TV stable.* Let j^* be the number of the query output by M^*. Since stable algorithms generalize (Theorem 12), we have

$$\mathbb{E}\left(q_{j^*}(\mathbf{s}) - q_{j^*}(\mathcal{D})\right) \le \epsilon.$$

On the other hand, by the definition of j^*,

$$\max_{j} \left(a_j - q_j(\mathcal{D})\right) = a_j^* - q_{j^*}(\mathcal{D}) = \left(a_{j^*} - q_{j^*}(\mathbf{s})\right) + \left(q_{j^*}(\mathbf{s}) - q_{j^*}(\mathcal{D})\right) .$$

The first term on the right-hand side is bounded above by the empirical error, which is at most α in expectation, by assumption. The second term is the generalization error of M^*, which we just bounded. Taking expectations, we get the desired upper bound, namely:

$$\mathbb{E} \max_j \left(a_j - q_j(\mathcal{D})\right) \leq \alpha + \epsilon \,.$$

A symmetric argument shows that $\mathbb{E} \max_j \left(q_j(\mathcal{D}) - a_j\right) \leq \epsilon + \alpha$. ■

As a side note, the reduction from many rounds to two rounds via a "monitor" works even without an algorithm that is empirically accurate. It requires a slightly more complex monitor which is itself distributionally stable. We may return to the proof later in the class. For now we state only the result.

Lemma 16 (Generalization of TV-Stable Mechanisms over Many Rounds) *If M is ϵ-TV stable then, for every distribution $\mathcal{D}$, and for every analyst A, when $\mathbf{s} \sim \mathcal{D}^n$, the expected maximum generalization error of the mechanism is*

$$\mathop{\mathbb{E}}_{\substack{\mathbf{s}\sim\mathcal{D}^n \\ \textit{coins of } M,A}} \left(\max_{j=1}^{k} |q_j(\mathbf{s}) - q_j(\mathcal{D})| \right) \leq \epsilon + \sqrt{\frac{\log k}{n}} \,.$$

We will see other uses of the monitor argument later in the course.

4 Composition of distributional notions

We now turn to designing stable algorithms that answer many queries. To do so, we need a composition statement analogous to what we had for compressible algorithms. What happens when we chain together several algorithms that are each distributionally stable? Do stability parameters add up?

Suppose we have an interactive mechanism M that interacts with an analyst over k rounds. We can break it into k spearate mechanisms $M_1, M_2, ...M_k$, where each of the mechanisms takes as input the original data set $\mathbf{s}$, as well as an internal state (denoted $state_i$), and the query q_i. The output now consists of the answer a_i and the updated internal state state_{i+1}. We call M the *adaptive sequential composition* of $M_1, ..., M_k$ (where "adaptive" comes from the fact that the algorithms' behavior depends on outcomes of previous rounds).

Theorem 17 (Distributional stability notions compose adaptively) *Suppose that $M = (M_1, ..., M_k)$.*

1. *Suppose each M_i is ϵ-TV stable, that is: for every value $state_i$, the randomized map $M_i(\cdot, state_i)$ (which maps $\mathbf{s}$ to $(a_i, state_{i+1})$) is ϵ-TV stable. Then, for every analyst A, the interactive process $A \circ M$ is $k\epsilon$-TV stable.*
2. *Suppose that $M = (M_1, ..., M_k)$ and each M_i is τ-KL stable. Then, for every analyst A, the interactive process $A \circ M$ is $k\tau$-KL stable.*
3. *Suppose that $M = (M_1, ..., M_k)$ and each M_i is ϵ-differentially private ($d_\diamond$-stable). Then, for every analyst A, the interactive process $A \circ M$ is $k\epsilon$-differentially private.*

Proof As with our proof of generalization, using closure under postprocessing, we can reduce this lemma to a simpler statement with no explicit analyst. Consider a sequence of mechanisms $M_1, ..., M_k$, where M_i takes $\mathbf{s}$ and the outputs $o_1, ..., o_{i-1}$ of previous mechanisms as input and produces output o_i. Now consider the joint output $(M_1, ..., M_k)$ obtained by running the mechanisms sequentially.

We prove part (2) of the theorem. We leave parts (1) and (3) as similar exercises.

Suppose that $M_i(\cdot, o_1, ..., o_{i-1})$ is τ-KL-stable for every setting of $o_1, ..., o_{i-1}$. To prove part (2) of the theorem, it is sufficient to prove that M is $k\tau$-KL stable. Fix data sets $\mathbf{s}, \mathbf{s}'$ that differ in one input. Let $Y_1, ..., Y_k$ be the (random) outputs of M on input $\mathbf{s}$, and $Z_1, ..., Z_K$ be the outputs of M on input $\mathbf{s}'$.

We can write the odds ration $\frac{\Pr(\mathbf{Y}=\mathbf{o})}{\Pr(\mathbf{Z}=\mathbf{o})}$ as a product $\prod_j \frac{\Pr\left(Y_j=o_j|Y_1^{j-1}=o_1^{j-1}\right)}{\Pr\left(Z_j=o_j|Z_1^{j-1}=o_1^{j-1}\right)}$. Thus, the log-odds ratio is a sum:

$$\ln \frac{\Pr(\mathbf{Y}=\mathbf{o})}{\Pr(\mathbf{Z}=\mathbf{o})} = \sum_{j=1}^{k} \ln \frac{\Pr\left(Y_j = o_j \mid Y_1^{j-1} = o_1^{j-1}\right)}{\Pr\left(Z_j = o_j \mid Z_1^{j-1} = o_1^{j-1}\right)} \,.$$

Taking the expectation over $\mathbf{o} \sim \mathbf{Y}$, we get

$$D_{KL}(Y\|Z) = \sum_{j=1}^{k} \mathbb{E}_{o_1^{j-1}\sim Y_1^{j-1}} \left(D_{KL}\Big(\underbrace{Y_j\big|_{Y_1^{j-1}=o_1^{j-1}}}_{M_j(o_1^{j-1},\mathbf{s})} \Big\| \underbrace{Z_j\big|_{Z_1^{j-1}=o_1^{j-1}}}_{M_j(o_1^{j-1},\mathbf{s'})} \Big) \right) \tag{10}$$

Equation (10) is an instance of a general phenomenon, the "chain rule" for KL divergence: given two distributions P, Q over pairs of elements, the divergence $D_{KL}(P\|Q)$ is the sum $D_{KL}(P_1\|Q_1) + \mathbb{E}_{x\sim P}\left(D_{KL}(P_{2,x}\|Q_{2,x})\right)$ where P_1, Q_1 are the marginal distributions of the first element of the pair under P and Q, respectively, and $P_{2,x}, Q_{2,x}$ are the conditional distributions on the second element conditioned on the first element being x.

Returning to our proof, the definition of stability says that every divergence on the right-hand side of (10) is at most τ, so the sum is at most $k\tau$. Hence, M is $k\tau$-KL stable. ■

Exercise 5 *Complete the proofs of parts 1 and 3 of the theorem.*

This theorem allows us to analyze complex algorithms and interactive processes modularly.

5 The Gaussian Mechanism

We now turn to our first real application of the machinery we've developed for distributionally stable algorithms.

Theorem 18 *The Gaussian mechanism with $\sigma = \frac{\sqrt[4]{k}}{\sqrt{n}\sqrt[4]{\log k}}$ allows answering k adaptively selected statistical queries with expected error*

$$O\left(\frac{\sqrt[4]{k}\log k}{\sqrt{n}}\right).$$

Proof We know that one iteration of the Gaussian mechanism is $\frac{1}{2n^2\sigma^2}$-KL stable. Using the theorem on adaptive sequential composition, we see that the composed mechanism is $\frac{k}{2n^2\sigma^2}$-KL stable, and hence $\frac{\sqrt{k}}{2n\sigma}$-TV stable (by Pinsker's inequality). On the other hand, the expected maximum *empirical* error of the Gaussian mechanism on a sequence of k queries is $O(\sigma\sqrt{\log k})$, since the probability that each individual query deviates by more that $c\sigma\sqrt{\log k}$ is $e^{-\Omega(c)}/k$. The overall expected population error is thus

$$O\left(\frac{\sqrt{k}}{n\sigma} + \sigma\sqrt{\log k}\right).$$

Setting $\sigma = \frac{\sqrt[4]{k}}{\sqrt{n}\sqrt[4]{\log k}}$ yields the desired result. ■

Note that we could not have gotten this result using only the composition results for TV stability, since those parameters would add up on the "wrong scale". The Gaussian noise mechanism for a single query is $\frac{1}{2n\sigma}$-TV stable. Applying composition for TV stability to the k-query mechanism shows that the k-query version is $\frac{k}{2n\sigma}$-TV stable, which is much worse than the $\frac{\sqrt{k}}{2n\sigma}$ bound one gets via KL stability.

The advantage of working with distributional stability instead of compression bounds is we get much tighter composition guarantees, often yielding a quadratic improvement in utility. The disadvantage is that distributionally stable algorithms are more complicated, since they involve extra randomization, and trickier to analyze.

At this point, it is worth asking if one can do can get significantly better error than the bound of Theorem 18. What is the best possible guarantee for the Gaussian mechanism? Is there any mechanism that can achieve better guarantees for arbitrary sequences of adaptive queries? We will see a complete answer to the first, and a partial answer to the second, in the coming lectures.

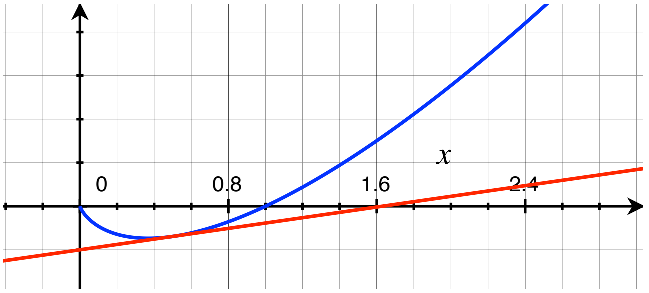


Figure 2: The function $f(x) = x\ln(x)$ is strictly convex on $(0, \infty)$. The red line shows the linear lower bound at $x = 0.5$.

6 Working with divergences

We now tie up our remaining loose ends by proving the required properties of the KL divergence (and introducing some useful inequalities along the way).

6.1 Jensen's inequality

Recall from earlier lectures that a set $C \subseteq \mathbb{R}^d$ is convex if for every two points $x, y \in C$, the line segment $\bar{x}y$ is contained in C. A function $f : C \to \mathbb{R}$ is convex on C if, at every value x, there is a linear function tangent to f at x that bounds f below on the whole domain. That is, there exists $u \in \mathbb{R}^d$ such that for every $y \in C$, $f(y) \geq f(x) + \langle u, y - x\rangle$. (If f is differentiable, then u is unique and is the gradient $\nabla f(x)$; otherwise, the set of vectors u that fit the condition is called the *subgradient set* of f at x, and denoted $\partial f(x)$). See Figure 6.1.

We say a function is *strictly convex* if the linear lower bound is strict everywhere except at x. That is, for every $y \in C \setminus \{x\}$, we have $f(y) > f(x) + \langle u, y - x\rangle$.

If a function is twice differentiable on its domain C, then it is strictly convex if and only if its second derivative is positive (or positive definite, in dimension greater than 1) on all of C.

Exercise 6 *Prove that $f(x) = x\ln(x)$ and $f(x) = \ln(1/x)$ are both strictly convex on $(0, \infty)$.*

Lemma 19 (Jensen's inequality) *Let f be a convex function on $C \subseteq \mathbb{R}^d$ and X a random variable taking values in C with finite expectation. Then*

$$f(\mathbb{E}(X)) \leq \mathbb{E}(f(X)).$$

Furthermore, if f is strictly convex, then we have equality if and only if X is constant (that is, X equals some particular value with probability 1).

Proof Let $\mu = \mathbb{E}(X)$ and $f(\mu) + \langle u_\mu, y - \mu\rangle$ be a linear lower bound to f that is tangent to f at μ. Then

$$\mathbb{E}(f(X)) \underbrace{\geq}_{\text{convexity}} \mathbb{E}(f(\mu) + \langle u_\mu, X - \mu\rangle) \underbrace{=}_{\substack{\text{linearity}\\ \text{of expectation}}} f(\mu) + \langle u_\mu, \underbrace{\mathbb{E}(X) - \mu}_{0}\rangle = f(\mu).$$

When f is strictly convex, the first inequality be an equality only when the random variable X places 0 probability mass on the set $C \setminus \{\mu\}$ (since on that set, $f(y) > f(\mu) + \langle u_\mu, y - \mu\rangle$). Thus, the inequality is tight only when $X = \mu$ with probability 1. ■

Using Jensen's inequality, one can prove the following well known inequality, whose proof is left as an exercise:

Lemma 20 (Log-Sum Inequality) *Let $a_1, \dots, a_n$ and $b_1, \dots, b_n$ be nonnegative numbers. Denote the sum of all a_i's by a and the sum of all b_i's by b. Then*

$$\sum_{i=1}^{n} \frac{a_i}{a} \log \frac{a_i}{b_i} \geq \log \frac{a}{b}$$

with equality if and only if $\frac{a_i}{b_i}$ are equal for all i.

6.2 Basic Properties of KL

We collect here a few properties of KL divergence that we used when discussing stability.

Lemma 21 *For every two distributions P and Q on a set $\mathcal{X}$:*

1. *$D_{KL}(P\|Q) \geq 0$ with equality if and only if $P = Q$.*
2. *If $d_\diamond(P,Q) = \epsilon$, then $D_{KL}(P\|Q) \leq D_{KL}(P\mathcal{Q}) + D_{KL}(Q\|P) \leq \epsilon(e^\epsilon - 1)$.*
3. *(Chain rule for KL) If $\mathcal{X}$ is a product of two sets $\mathcal{X}_1 \times \mathcal{X}_2$, (so that P, Q are distributions over pairs), the divergence $D_{KL}(P\|Q)$ is the sum $D_{KL}(P_1\|Q_1) + \mathbb{E}_{x\sim P}\left(D_{KL}(P_{2,x}\|Q_{2,x})\right)$ where P_1, Q_1 are the marginal distributions of the first element of the pair under P and Q, respectively, and $P_{2,x}, Q_{2,x}$ are the conditional distributions on the second element conditioned on the first element being x.*
4. *(Monotonicity under postprocessing) For every randomized map A taking inputs in $\mathcal{X}$, $D_{KL}(A(P)\|A(Q)) \leq D_{KL}(P\|Q)$.*
5. *(Pinsker's inequality) $d_{TV}(P,Q) \leq \sqrt{\frac{1}{2} D_{KL}(P\|Q)}$*

Proof

1. $D_{KL}(P\|Q) = \mathbb{E}_{X\sim P}\left(\ln \frac{P(X)}{Q(X)}\right) = \mathbb{E}_{X\sim P}\left(-\ln \frac{Q(X)}{P(X)}\right)$. Applying Jensen's inequality to $f(x) = -\ln(x)$, we get $D_{KL}(P\|Q) \geq -\ln\left(\mathbb{E}_{X\sim P}\left(\frac{Q(X)}{P(X)}\right)\right)$. If we expand the definition of expectation, we see that the denominator cancels in the expression $\mathbb{E}_{X\sim P}\left(\frac{Q(X)}{P(X)}\right)$, and we have $\mathbb{E}_{X\sim P}\left(\frac{Q(X)}{P(X)}\right) = \int_{x\in supp(P)} \frac{Q(x)}{P(x)} P(x)dx = \int_{x\in supp(P)} Q(x) \leq 1$ where *supp*(P) is the set of points with positive density (or mass, in the discrete case). Thus $D_{KL}(P\|Q) \geq 0$. We get equality in Jensen's inequality if and only if $\frac{Q(X)}{P(X)}$ is constant (since $-\ln(\cdot)$ is strictly convex). Since P and Q both integrate to 1, we thus get equality if and only if $P = Q$.
2. By nonnegativity of KL, we have $D_{KL}(P\|Q) \leq D_{KL}(P\mathcal{Q}) + D_{KL}(Q\|P)$. Expanding the sum, we have

$$\begin{aligned} D_{KL}(P\mathcal{Q}) + D_{KL}(Q\|P) &= \mathop{\mathbb{E}}_{X\sim P}\left(\ln \tfrac{P(X)}{Q(X)}\right) + \mathop{\mathbb{E}}_{X\sim Q}\left(\ln \tfrac{Q(X)}{P(X)}\right) \\ &= \mathop{\mathbb{E}}_{X\sim P}\left(\ln \tfrac{P(X)}{Q(X)} + \tfrac{Q(X)}{P(X)} \ln \tfrac{Q(X)}{P(X)}\right) = \mathop{\mathbb{E}}_{X\sim P}\left(1 - \tfrac{Q(X)}{P(X)}\right) \ln \tfrac{P(X)}{Q(X)} \\ &\leq \mathop{\mathbb{E}}_{X\sim P}\left(\left|1 - \tfrac{Q(X)}{P(X)}\right| \cdot \left|\ln \tfrac{P(X)}{Q(X)}\right|\right). \end{aligned}$$

   Now we can use the fact that $d_\diamond(P,Q) = \epsilon$: the term $\left|\ln \frac{P(X)}{Q(X)}\right|$ in the last expression is always at most ϵ, and the term $\left|1 - \frac{Q(X)}{P(X)}\right|$ is at most $\max(1 - e^{-\epsilon}, e^\epsilon - 1) = e^\epsilon - 1$. The expectation on the right-hand side is thus at most $\epsilon(e^\epsilon - 1)$.
3. This is the chain rule for D_{KL}, which we proved when proving the composition lemma. We include it in this lemma just to have important properties of KL collected in one place.
4. First, note that if X and Y are distributed according to P and Q respectively, then the KL divergence between the distributions of the pairs $X, A(X)$ and $Y, A(Y)$ is exactly the same as $D_{KL}(P\|Q)$:

$$D_{KL}\big((X, A(X)), (Y, A(Y))\big) = \mathop{\mathbb{E}}_{\substack{x\sim P \\ z\sim A(X)}}\left(\ln \frac{P(x)\Pr(z = A(x))}{Q(x)\Pr(z = A(x))}\right) = \mathop{\mathbb{E}}_{\substack{x\sim P \\ z\sim A(X)}}\left(\ln \frac{P(x)}{Q(x)}\right) = D_{KL}(P\|Q)\,.$$

We can now apply the chain rule:

$$D_{KL}\big((X,A(X)),(Y,A(Y))\big) = D_{KL}(A(X),A(Y)) + \mathop{\mathbb{E}}_{\substack{x\sim P\\ z\sim A(x)}}\left(\underbrace{D_{KL}\Big(X\big|_{z=A(X)}\big\|Y\big|_{z=A(Y)}\Big)}_{\geq 0}\right).$$

By the nonnegativity of KL divergence, the expectation on the right-hand side is always nonnegative, so we get

$$D_{KL}(A(X),A(Y)) \leq D_{KL}\big((X,A(X)),(Y,A(Y))\big) \leq D_{KL}(X\|Y)\,,$$

as desired.

5. Recall that $d_{TV}(P,Q) = \sup_E |P(E)-Q(E)|$. Let E^* be a fixed event, and let P',Q' be distributions on $\{0,1\}$ with $P'(0) = P(E) \overset{\text{def}}{=} p$ and $Q'(0) = Q(E) \overset{\text{def}}{=} q$. Let $D_{KL}(p\|q)$ denote $D_{KL}(P'\|Q') \leq D_{KL}(P\|Q)$ (since processing can only decrease divergence). Our inequality reduces to:

   $$D_{KL}(p\|q) - 2(p-q)^2 \geq 0\,.$$

   To prove this, fix an arbitrary $p \in [0,1]$ and compute the partial derivative with respect to q:

   $$\frac{\partial}{\partial q}\big(D_{KL}(p\|q) - 2(p-1)^2\big) = \frac{q-p}{q(1-q)} - 4(q-p) = (q-p)\Big(\tfrac{1}{q(1-q)} - 4\Big)\,.$$

   Now $\frac{1}{q(1-q)} - 4$ is never negative (since $q(1-q) \leq 4$), so the function $g(q) = D_{KL}(p\|q) - 2(p-1)^2$ is increasing for $q > p$ and decreasing for $q < p$. Thus the minimum occurs at $q = p$, where the function is 0.

■

7 Notes

The stability-based approach to analyzing generalization error dates back to work of Devroye and Wagner [DW79]. The topic is now well studied in learning theory–see, for example, Bousquet and Elisseeff [BE02] and Shalev-Shwartz et al. [SSSSS10] for thorough treatments and further references.

The idea of using distributional stability in the context of adaptive statistical queries comes from Dwork et al. [DFH$^+$15]. Theorem 11 was folklore in the differential privacy literature for some time. The first application we are aware of is in [BST14], which used the lemma to relate the empirical error and generalization error of differentially private learning algorithms.

The presentation of the "monitor" argument comes from Bassily et al. [BNS$^+$16]. The analysis of the Gaussian mechanism via KL stability is implicit in [BNS$^+$16] but draws on ideas in Russo and Zou [RZ16] and Wang et al. [WLF16].

References

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[BNS$^+$16] Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1046–1059. ACM, 2016.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 464–473, 2014.

[DFH$^+$15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *STOC*, pages 117–126. ACM, 2015.

[DW79] L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979.

[RZ16] Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *19th International Conference on Artificial Intelligence and Statistics*, pages 1232–1240, 2016. arXiv:1511.05219.

[SSSSS10] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *JMLR*, 2010.

[WLF16] Yu-Xiang Wang, Jing Lei, and Stephen E Fienberg. A minimax theory for adaptive data analysis. arXiv:1602.04287 [stat.ML], 2016.