The Algorithmic Foundations of Adaptive Data Analysis	November 27, 2017
Lecture 17-18	
Lecturer: Aaron Roth	*Scribe: Aaron Roth*

Compression Schemes

In this lecture, we give another way to argue about the generalization properties of learning algorithms in adaptive settings. The technique will be based on *compression*, but it will be distinct from transcript compressibility that we studied earlier in the course. In this lecture, we will not ask for compressibility to a small number of *bits*. Instead, we will ask that the transcript of an interaction should be reconstructible from a small number of *data points* in our empirical sample S. This is different, because in very large (or infinite) data domains, it may not be possible to describe individual samples with a small number of bits. An implication of this connection to compression schemes will be that there are some things that can be learned in a generalizable way, even in adaptive settings, that cannot be learned subject to differential privacy. This separation suggests that differential privacy is not the whole story when it comes to adaptive data analysis.

It might at first seem counter-intuitive that it is possible to represent an accurately learned hypothesis with a small number of training examples, but as it turns out, for many learning problems, not only is this possible, but it leads to nearly optimal sample complexity bounds. Our plan for this lecture will be to first define and prove the generalization properties of sample compression schemes, and then derive sample compression schemes for a general learning problem. Finally, we will provide an adaptive composition theorem and explain the relevance to adaptive data analysis.

1 Compression Schemes and Generalization

To get an intuition for compression bounds, we start with a warmup. We will talk about a learning setting in which there is some distribution $\mathcal{P}$ over a set of labelled examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$. We leave the label space abstract, since we will be able to speak of multi-class classification and regression as well, but for intuition, you can think of your favorite space of labelled examples $\mathcal{X} \times \mathcal{Y} = \{0,1\}^d \times \{0,1\}$. A hypothesis is just a function mapping features to labels $h : \mathcal{X} \to \mathcal{Y}$. To evaluate a hypothesis, we define a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$. For example, a popular choice is classification error: $\ell(h(x), y) = \mathbb{1}[h(x) \neq y]$. Given a distribution $\mathcal{P}$ and a classifier h, write $L_{\mathcal{P}}[h] = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(h(x), y)]$, and given a dataset S, write $L_S[h] = \frac{1}{|S|}\sum_{(x,y)\in S} \ell(h(x), y)$.

Now, imagine that we sample two independent datasets: $S \sim \mathcal{P}^k, T \sim \mathcal{P}^{n-k}$. Suppose we construct some example h_S from S, but then evaluate h_S on T. Since S and T are independent, $L_T(h_S)$ is an unbiased estimator for $L_{\mathcal{P}}(h_S)$. We can get high probability bounds on the error of our estimate of $L_{\mathcal{P}}(h_S)$, as evaluated on T using a Chernoff bound. By using a variant (Bernstein's inequality), we can get a slightly improved bound:

$$\Pr\left[|L_{\mathcal{P}}(h_S) - L_T(h_S)| \geq \sqrt{\frac{2L_T(h_S)\log(1/\delta)}{|T|}} + \frac{4\log(1/\delta)}{|T|}\right] \leq \delta \tag{1}$$

All that is used to get a bound like this is that S and T are independent of each other. We could have simply sampled a single dataset $D \sim \mathcal{P}^n$, and defined S to be the first k indices of D, and defined T to be the rest. And there is nothing special about the first k indices — so long as S is defined to be any set of k indices of D, fixed before the data is drawn, the above bound is valid. But note — if this bound holds for every fixed set of k indices, then we can simply union bound over the $< n^k$ possible choices of k fixed indices, to obtain the following theorem:

Theorem 1 *Let k be any integer, and let $B : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{H}$ be any mapping from sequences of k examples to classifiers in $\mathcal{H}$. Let let $A : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}$ be a learning rule that takes as input a dataset $S = ((x, y)_1, \ldots, (x, y)_n)$, and outputs a classifier $h \in \mathcal{H}$ such that $A(S) = B((x, y)_{i_1}, \ldots, (x, y)_{i_k})$ for*

some set of indices $(i_1, \ldots, i_k) \in [n]^k$. Then for any distribution $\mathcal{P}$ with probability at least $1-\delta$ over the draw of $S \sim \mathcal{P}^n$, we have:

$$|L_{\mathcal{P}}[A(S)] - L_S(A(S))| \leq \sqrt{\frac{2L_S(A(S)4k\log(n/\delta))}{|S|}} + \frac{8k\log(m/\delta)}{|S|} + \frac{2k}{n}$$

Proof Let $V = \{(x,y)_j : j \notin \{i_1, \ldots, i_k\}\}$ be the set of examples not chosen by A in selecting the hypothesis $A(S)$. For any set of indices $I \in [n]^k$, let $h_I = B(\{(x,y)_i : i \in I\})$ be the hypothesis induced by this set of data points. Then we have:

$$\Pr\left[|L_{\mathcal{P}}[A(S)] - L_V(A(S))| \geq \sqrt{\frac{2L_V(A(S))\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}\right] \leq$$

$$\Pr\left[\exists I \in [n]^k : |L_{\mathcal{P}}[h_I] - L_V(h_I)| \geq \sqrt{\frac{2L_V(h_I)\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}\right] \leq n^k\delta$$

where the final inequality follows from the bound we proved in Equation 1, and a union bound over all n^k choices of I. Setting $\delta = \delta/n^k$, and the fact that $|V| \geq |S|/2$ yields the bound:

$$\Pr\left[|L_{\mathcal{P}}[A(S)] - L_V(A(S))| \geq \sqrt{\frac{4L_V(A(S))k\log(n/\delta)}{|V|}} + \frac{8k\log(n/\delta)}{|V|}\right] \leq \delta$$

Finally, we observe by the triangle inequality that:

$$\begin{aligned}|L_{\mathcal{P}}[A(S)] - L_S(A(S))| &\leq |L_{\mathcal{P}}[A(S)] - L_V(A(S))| \\ &+ \left|\frac{|V|}{|S|}L_V(A(S)) - L_S(A(S))\right| + \left|\frac{|V|}{|S|}L_V(A(S)) - L_V(A(S))\right| \\ &\leq |L_{\mathcal{P}}[A(S)] - L_V(A(S))| + \frac{k}{n} + \frac{k}{n}\end{aligned}$$

where the final inequality holds because $|S \setminus V| = \frac{k}{n}|S|$ The claimed bound follows. ■

We can generalize this somewhat. Suppose our algorithm B was not constrained to output a classifer, but an element of an abstract set $\mathcal{Y}$. As in Lecture 4 (when we analyzed the generalization properties of compressible learners), suppose we have a map R that associates with each $y \in \mathcal{Y}$ a set of $R(y) \subseteq \mathcal{X}^{n-k}$ of possible values for the points that were not used in the output (corresponding to indices in T (the set V in the previous proof). Given an output $Y = A(\mathbf{S})$, we can ask what is the probability that V lies in $R(Y)$, and how that probability compares to the chance that a fresh, independent sample of $n-k$ points from $\mathcal{P}$ would lie in $R(Y)$.

Theorem 2 *Let k be any integer, and let $B : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{H}$ be any mapping from sequences of k examples to classifiers in $\mathcal{H}$. Let let $A : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}$ be a learning rule that takes as input a dataset $S = ((x,y)_1, \ldots, (x,y)_n)$, and produces an output in $\mathcal{Y}$ such that $A(S) = B((x,y)_{i_1}, \ldots, (x,y)_{i_k})$ for some set of indices $(i_1, \ldots, i_k) \in [n]^k$. Let R be an arbitrary map that associates each potential output $y \in \mathcal{Y}$ with an event $R(y) \subseteq \mathcal{X}^{n-k}$. Then for any distribution $\mathcal{P}$,*

$$\mathop{\mathbb{E}}_{\substack{\mathbf{S}\sim\mathcal{P}^n \\ Y = A(\mathbf{S})}} \frac{\Pr(\mathbf{S} \in R(Y))}{\Pr_{\mathbf{S}'\sim\mathcal{P}^n}(\mathbf{S}' \in R(Y))} \leq \binom{n}{k}.$$

We can use this theorem to recover the previous one by setting $R(y)$ to be the set of data sets V for which $L_V[y]$ differs significantly from $L_{\mathcal{P}}[y]$ (where "significantly" is chosen to make the probability of the deviation at most δ).

This motivates the definition of a *compression scheme* — a kind of learning algorithm that produces compressible hypotheses, of the sort that we just argued do not overfit. We will start by defining a compression scheme in the "realizable" setting — i.e. the setting in which there is always some $h \in \mathcal{H}$ that perfectly labels the data.

Definition 3 (Realizable Compression Schemes) *We say that $\mathcal{H}$ has a realizable compression scheme of size k if: for all n, there exists a pair of algorithms $A : (\mathcal{X} \times \mathcal{Y})^n \to [n]^k$, $B : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{H}$ such that for all $h \in \mathcal{H}$, if $S = ((x_1, h(x_1)), \ldots, (x_n, h(x_n)))$ and $I = A(S)$, then $h' = B(S_I)$ satisfies $L_S(h') = 0$. Here $S_I = \{(x_i, y_i) : i \in I\}$.*

In other words, we can perfectly "compress" all of the labels in S to a small set I of indices, and then successfully decompress them with the algorithm B.

Of course, in real life, we rarely have a dataset that can be perfectly labeled by any $h \in \mathcal{H}$. So we can talk about compression schemes in the non-realizable case as well:

Definition 4 (Nonrealizable Compression Schemes) *We say that $\mathcal{H}$ has a compression scheme of size k if: for all n, there exists a pair of algorithms $A : (\mathcal{X} \times \mathcal{Y})^n \to [n]^k$, $B : (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{H}$ such that if $S = ((x_1, y), \ldots, (x_n, y))$ and $I = A(S)$, then $h' = B(S_I)$ satisfies $L_S(h') \leq L_S(h)$ for all $h \in \mathcal{H}$.*

It isn't hard to argue that if a class $\mathcal{H}$ has a compression scheme in the realizable case of size k, then it has one in the non-realizable case as well:

Lemma 5 *For any class $\mathcal{H}$, if $\mathcal{H}$ has a realizable compression scheme of size k, then it has a non-realizable compression scheme of size k.*

The proof is left as an exercise.

2 Classes admitting Compression Schemes

2.1 Linear Threshold Functions

So we know that if we can learn with a compression scheme of size k, we can safely generalize: If we want to have generalization error $O(\epsilon)$, it suffices to take a sample of size $|S| \geq \tilde{O}(k/\epsilon^2)$ in the non-realizable case, and a sample of size $|S| \geq \tilde{O}(k/\epsilon)$ in the realizable case. It turns out many learners have optimal compression schemes. Here we go through one example: the class $\mathcal{H}$ of linear threshold functions. These are the kinds of functions learned by logistic regression, SVMs, etc.

Let $\mathcal{X} = \mathbb{R}^d$, let $\mathcal{Y} = \{-1, 1\}$, and let $\ell(y, y') = \mathbb{1}(y \neq y')$. Define the class of functions:

$$\mathcal{H}_{LTF} = \{h(x) \doteq \text{sign}(w \cdot x) : w \in \mathbb{R}^d\}$$

Here, $\text{sign}(w \cdot x) = 1$ if $w \cdot x > 0$ and $\text{sign}(w \cdot x) = -1$ otherwise.

Theorem 6 *$\mathcal{H}_{LTF}$ has a compression scheme of size $d + 1$.*

Remark Its actually not hard to argue that it has a compression scheme of size d — this is left as an exercise.

Proof By our earlier observation, we only need to argue that $\mathcal{H}_{LTF}$ has a *realizable* compression scheme of size $d + 1$. Fix any $h \in \mathcal{H}_{LTF}$ and any dataset $S = \{(x_1, h(x_1)), \ldots, (x_n, h(x_n))\}$. We will alternately write y_i for $h_i(x_i)$, to emphasize that we don't know what h is. First, we define our compression algorithm A. $A(S)$ operates as follows:

1. Define $S^+ = \{x_1 y_1, \ldots, x_n y_n\} \subset \mathbb{R}^d$.
2. Let C be the convex hull of S^+.
3. Let $w^* = \arg\min_{w \in C} ||w||$.
4. Write w^* as the convex combination of at most $d + 1$ points in S^+. Output I, the index set of these points.

Our decompression scheme $B(S_I)$ will be as follows:

1. Define $S_I^+ = \{x_i y_i : i \in I\}$.

2. Let C' be the convex hull of S_I^+.

3. Let $w' = \arg\min_{w \in C'} ||w||$.

4. Output $h'(x) \doteq \text{sign}(w' \cdot x)$.

We have a bunch of things to prove. First, step 4 of algorithm A seems tricky — why can we write w^* as a convex combination of $d+1$ points in S^*? This is a consequence of Carathéodory's Theorem:

Lemma 7 (Carathéodory's Theorem) *If a point $x \in \mathbb{R}^d$ lies in the convex hull of a set P, then it can be written as a convex combination of at most $d+1$ points in P.*

We will defer the proof of this for now.

Next, we need to show that if we run this compression scheme A and then decompress, the hypothesis h' output by B will recover all of the labels in S perfectly. First we show a simple lemma stating that the origin does not appear in our convex hull C.

Lemma 8 $0 \notin C$

Proof Note that since we have $y_i = h(x_i) = \textbf{sign}(w \cdot x_i)$ for some w, it must be that $w \cdot y_i x_i > 0$ for all i. Consider any point $x \in C$. By definition, there are non-negative numbers α_i summing to 1 such that $x = \sum_i \alpha_i x_i y_i$. Thus:

$$w \cdot x = \sum_i \alpha_i w \cdot x_i y_i > 0$$

So it must be that $x \neq 0$.

■ We now claim that $h^*(x) \doteq \text{sign}(w^* \cdot x)$ is such that for all i, $h^*(x_i) = h(x_i)$, or equivalently, for all i, $w^* \cdot x_i y_i > 0$. (It can be shown that w^* is actually the *maximum margin* linear classifier. Suppose otherwise – that for some x_i, $w^* \cdot x_i y_i \leq 0$. Consider the point $\tilde{w} = \alpha \cdot w^* + (1 - \alpha x_i y_i)$, for

$$\alpha = \frac{||w^*||^2}{||w^*||^2 + ||x_i||^2}$$

Then we have:

$$\begin{aligned}
||\tilde{w}||^2 &= (1-\alpha)^2||w^*||^2 + \alpha^2||x_i||^2 + 2\alpha * (1-\alpha)\tilde{w} \cdot x_i y_i \\
&\leq (1-\alpha)^2||w^*||^2 + \alpha^2||x_i||^2 \\
&= \frac{||x_i||^4||w^*||^2 + ||w^*||^4||x_i||^2}{(||w^*||^2 + ||x_i||^2)^2} \\
&= \frac{||x_i||^2||w^*||^2}{||w^*||^2 + ||x_i||^2} \\
&= ||w^*||^2 \cdot \frac{||x_i||^2}{||w^*||^2 + ||x_i||^2} \\
&< ||w^*||^2
\end{aligned}$$

Here, the first inequality comes from the assumption that $w^* \cdot x_i y_i \leq 0$, and the second follows from Lemma 8, which implies that $||x_i||^2 > 0$ for all $x_i \in C$ But note that this contradicts the fact that $w \in C$, and so this contradicts the fact that w^* is the vector with smallest norm in C.

Great – we're almost done. We know that h^* reconstructs every label in S. All that remains to show is that the hypothesis h' output by the decompression B is such that $h' = h^*$. To do this, we just need to show that $w' = w^*$. But this is almost immediate. Since w^* can be written as a convex combination of the $d+1$ points in S_I^+, it must be that $w^* \in C'$. And $w' = \arg\min_{w \in C'} ||w||$, so $||w'|| \leq ||w^*||$. But $C' \subseteq C$, so we also have that $||w^*|| \leq ||w'||$, and so $||w^*|| = ||w'||$. But since the ℓ_2 norm is strongly convex, and C is convex, the minimization problem $\arg\min_{w \in C} ||w||$ has a unique solution, and so $w' = w^*$, which completes the proof. ■

All that is left is to prove Carathéodory's Theorem.

Proof [Carathéodory's Theorem]

Suppose x lies in the convex hull of a set of points C. This means that for some k, there exist points $x_1, \ldots, x_k \in P$ and numbers $\lambda_j \geq 0$, $\sum_{j=1}^{k} \lambda_j = 1$ such that:

$$x = \sum_{j=1}^{k} \lambda_j x_j$$

If $k \leq d+1$, we are done, so we can assume $k > d+1$. The argument will proceed by iteratively re-writing x as a convex combination of one fewer point, thereby decreasing k by 1, until we have $k \leq d+1$.

Consider the points:

$$x_2 - x_1, \quad x_3 - x_1, \quad \ldots \quad , x_k - x_1$$

Since this is a set of $> d$ points in a d dimensional space, they must be linearly dependent. This menas we can find scalars $\mu_2, \ldots, \mu_k$ not all zero such that:

$$\sum_{j=2}^{k} \mu_j (x_j - x_1) = 0$$

Define $\mu_1 = -\sum_{j=2}^{k} \mu_j$. Note that this gives:

$$\sum_{j=1}^{k} \mu_j x_j = 0, \quad \sum_{j=1}^{k} \mu_j = 0$$

(in other words, $x_1, \ldots, x_k$ are *affinely* dependent.) Let $J = \{i \in \{1, \ldots, k\} : \mu_i > 0\}$. Note that J must be non-empty, since the μ_i are not all zero, and yet sum to zero. Not also that for any α, we can write:

$$x = \sum_{j=1}^{k} \lambda_j x_j - \alpha \sum_{j=1}^{k} \mu_j x_j = \sum_{j=1}^{k} (\lambda_j - \alpha \mu_j) x_j$$

We make a particular choice of α:

$$\alpha = \min_{j \in J} \frac{\lambda_j}{\mu_j} \doteq \frac{\lambda_{i^*}}{\mu_{i^*}}$$

Note that with this choice of α, we have for every j, $\alpha \mu_j \leq \frac{\lambda_j}{\mu_j} \cdot \mu_j = \lambda_j$, and so $\lambda_j - \alpha \mu_j \geq 0$. Note also that for index i^*, we have $\lambda_{i^*} - \alpha \mu_{i^*} = 0$. Finally, $\sum_{j=1}^{k} (\lambda_j - \alpha \mu_j) = 1$. So the coefficients $(\lambda_j - \alpha \mu_j)$ can also be used to represent x as a convex combination of points in P, but now as a combination of only $k-1$ points. The result follows. ■

2.2 Classes with Bounded VC-dimension

So we showed that linear threshold functions have optimal compression schemes. (n.b.: What do we mean by optimal? For binary classification, the sample complexity of learning over worst-case distributions is characterized by the "VC-Dimension" d of a hypothesis class, and the best generalization error achievable is equal to the bound we proved for compression schemes, but with the compression size k replaced with d. For linear threshold functions, the VC-dimension is d, equal to its compressibility bound. When we go beyond binary classification, VC-dimension bounds are no longer always optimal, and indeed, sometimes compression bounds can give better generalization bounds). Was this a coincidence? No. Via a boosting argument, [DMY16] have shown that every hypothesis class for binary classification has a compression scheme of size that is not much larger than its VC-dimension:

Theorem 9 ([DMY16]) *Let $\mathcal{H}$ be a class of binary-valued functions with VC-dimension d. Then $\mathcal{H}$ has a compression scheme of size k on datasets of size $|S| = n$ for:*

$$k(n) = O\left(d \log(n)(\log\log(n) + \log d)\right)$$

3 Compression Schemes in Adaptive Data Analysis

It remains to observe that we can use compression schemes to reason about generalization in adaptive settings. Recall, that in order to do this, we need to argue two things: that the generalization bounds are robust to *post-processing* and to *composition*. That is:

1. Postprocessing: By *thinking* about a compressed hypothesis, an adversary should not be able to come up with another hypothesis that overfits. Observe that generalization bounds that follow from e.g. VC-dimension arguments do not necessarily satisfy this property, since a hypothesis could encode in its low order bits the entire dataset. A post-processing of this hypothesis could extract that information, and construct a maximally overfitting query.

2. Composition: Any hypothesis that results from adaptively making queries and receiving compressed hypotheses should also be compressible.

We will see that compression schemes easily satisfy these two properties. The post-processing guarantee follows almost immediately from the definition of a compression scheme. We call any hypothesis h that can be written as $h = B \circ A(S)$, a composition of an algorithm A that compresses S to a subset of size k, and is then decompressed into a hypothesis by an algorithm B as k-compressible.

Claim 10 *For any function $f : \mathcal{H} \to \mathcal{H}'$, and any k-compressible hypothesis $h \in \mathcal{H}$, $h' = f(h)$ is also k-compressible.*

Proof By assumption, h' can be written as $h' = f \circ B \circ A(S)$. Define $B' = f \circ B$. Then $h' = B' \circ A$, as required in the definition of k-compressibility. ■

Composition is also straight forward.

Claim 11 *For any m, let*

$$h_1 = M_1(S), h_2 = M_2(S; h_1), h_3 = M_3(S; h_1, h_2), \ldots, h_m = M_m(S; h_1, \ldots, h_{m-1})$$

such that for every i and every $h'_1, \ldots, h'_{i-1}$, $h_i = M_i(\cdot; h'_1, \ldots, h'_{i-1})$ is k_i-compressible. Then h_m is k compressible for $k = \sum_i k_i$.

Proof Fixing $h_1, \ldots, h_{i-1}$, write $h_i = M_i(\cdot; h'_1, \ldots, h'_{i-1}) = A_i \circ B_i(S)$, where $B_i(S)$ outputs $S_i \subset S$ of size k_i. Note that h_i is a post-processing of S_i, and so $S_1, \ldots, S_i$ uniquely determine S_{i+1}. Thus, the entire sequence of hypotheses $S_1, \ldots, S_m$ can be reconstructed from the sequence $(S_1, \ldots, S_m)$, via a single algorithm B. Thus we can view the entire adaptive composition as a single compression scheme of size $|S_1 \cup \ldots \cup S_m| \leq \sum_i k_i$. ■

Finally, we note that the ability to argue about generalization via compression schemes seemingly gives us a way of giving guarantees for certain kinds of adaptive learning problems more straightforwardly than either transcript (i.e. bit-length) compression or differential privacy do.

Consider the following very simple class $\mathcal{H}$ defined over the real interval $\mathcal{X} = [0, 1]$: $\mathcal{H}_T = \{h_t(x) \doteq \mathbb{1}(x \leq t) : t \in [0, 1]\}$. Each function $h_t \in \mathcal{H}_T$ is parameterized by a real valued point t in the unit interval, and simply identifies whether its input point x lies to the left or right of t. Although this is an extremely simple class of functions, there is generally no finite description of a function $h_t \in \mathcal{H}_T$: Specifying h_t requires specifying t, which can be an arbitrary real number. Despite this fact, it is easy to learn with respect to $\mathcal{H}_T$. It is straightforward to show e.g. that the VC-dimension of $\mathcal{H}_T$ is 1, but it is equally easy to give a compression scheme of size $k = 1$ for this class:

Claim 12 *$\mathcal{H}_T$ has a compression scheme of size $k = 1$.*

Proof Fix any dataset S labeled by a function $h_t \in \mathcal{H}_T$. Consider the following simple compression algorithm $A(S)$:

- Let $S^+ = \{(x_i, y_i) \in S : y = 1\}$

- Output the index $i^* = \arg\max x_i : (x_i, y_i) \in S^+$

The decompression algorithm B simply outputs $h \doteq h_{x_{i^*}}$. Note that by assumption, for any $(x_i, y_i) \in S$ with $y_i = 1$, $x_i \leq x_i^*$, and so $h(x_i) = 1$. Similarly, for any $(x_i, y_i) \in S$ with $y_i = 0$, $x_i > x_i^*$, and so $h(x_i) = 0$. Thus, $h = B \cdot A(S)$ perfectly recovers the labels, as required. ■

This compression scheme is what is called a *proper* learning algorithm: it outputs a hypothesis $h \in \mathcal{H}$ — i.e. it learns a hypothesis from the same class from which the dataset was labeled. But there is no proper learning algorithm for $\mathcal{H}_T$ that is differentially private, or that outputs a hypothesis of finite description length. We will prove the description length statement, which is simple, but simply quote the impossibility result for private learning.

Theorem 13 *Fix any finite subset $\mathcal{H}'_T \subseteq \mathcal{H}_T$. Consider any learning algorithm $M : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}'_T$. Then there is a distribution $\mathcal{D}$ over examples perfectly labeled by some $h \in \mathcal{H}$, but such that for $S \sim \mathcal{D}^n$, and $h' = M(S)$, $err(h, \mathcal{D}) = 1/2$.*

Proof Since $\mathcal{H}'_T$ is finite, we can write $\mathcal{H}'_T = \{h_{x_1}, \ldots, h_{x_m}\}$ such that $x_1 < \ldots < x_m$. Let y, z be any pair of points such that $x_1 < y < z < x_2$. Let $\mathcal{D}$ be the distribution that puts half of its probability mass on y and half on z. Let the target hypothesis be $h = h_y$. So, $h(y) = 1$ and $h(z) = 0$. But for any $h' \in \mathcal{H}'_T$, $h'(y) = h'(z)$, so the error of h' is $1/2$. ■

Theorem 14 ([BNSV15]) *Fix n, and any (ϵ, δ)-differentially private algorithm $M : \{[0,1] \times \{0,1\}\}^n \to \mathcal{H}_T$. Then if $\epsilon \leq 1/2$, $\delta \leq O(1/n^2)$, there is a distribution $\mathcal{D}$ such that for $S \sim \mathcal{D}^n$ and labelled with a function $h \in \mathcal{H}_T$, if $h' = M(S)$, with constant probability,* $\text{err}(h, \mathcal{D}) \geq 1/2$.

Bibliographic Information Our presentation of compression schemes follows that of [SSBD14]. [DMY16] provide an in depth study of compression schemes, including proving that every learnable binary hypothesis class has a nearly optimal compression scheme. The connection between compression schemes and adaptive data analysis was made in [CLN+16].

References

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 634–649. IEEE, 2015.

[CLN+16] Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In *Conference on Learning Theory*, pages 772–814, 2016.

[DMY16] Ofir David, Shay Moran, and Amir Yehudayoff. Supervised learning through the lens of compression. In *Advances in Neural Information Processing Systems*, pages 2784–2792, 2016.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.