COS 598E: Unsupervised Learning — Week 2

Lecturer: Elad Hazan — Scribe: Kiran Vodrahalli

Contents

1 Introduction

What I want to do today is first continue with some theoretical observations about unsupervised learning, and prove generalization bounds (non-generative) for unsupervised learning. This is as far as I can tell the only way to give generalization error. Last time we talked about unsupervised learning and philosophy, and we talked about a specific definition. We will come up with convex relaxations and efficient algorithms for solving the problem in this model.

2 A “Non-generative” Approach

Consider a hypothesis class composed of two functions $(f, g) \in \mathcal{H}$, one is a decoder and the other is an encoder. We measure reconstruction error by $\mathbf{E}_{x\sim D}[x - g \circ f(x)]$. If this value is zero, we didn't lose any information. We have D is an unknown distribution, like in the PAC model. A family of encoder-decoders is learnable if we can give a bound on the number of examples required to get generalization error which is small. We define the loss

$$L_D(h) = \mathbf{E}_{x\sim D}[|x - h(x)|] \tag{1}$$

Just like in PAC learning, we take a sample $S \sim D^m$ where $S = \{x_1, \cdots, x_m\}$, and we define sample loss $L_S(h) = \mathbf{E}_{x\sim S}[|x - h(x)|]$. It's learnable if there exists an (efficient) algorithm where after $m(\epsilon, \delta)$ samples it finds $\bar{h}$ s.t. with probability $1-\delta$, $L_D(\bar{h}) \leq L_D(h^*)+\epsilon$.

This view looks at learning as compression with low reconstruction error. We can sometimes add a term to the loss dependent on the number of bits required to express $f(x)$.

Now, how do we analyze PCA in this framework? What kind of guarantees can we hope to prove for PCA in terms of generalization?

Theorem 2.1. *PCA is learnable with sample complexity $\sqrt{\frac{d}{m}}$. This d can be improved to k (k-PCA), $x \in \mathbb{R}^d$, but we won't show this. Therefore, $m(\epsilon, \delta) \sim \frac{d}{\epsilon^2}$. We omit the $1/\delta$.*

Proof. Last time, we defined Rademacher complexity.

Definition 2.2. Rademacher complexity.
Consider a family of mappings $F : X \to \mathbb{R}$. The Rademacher complexity is

$$R_S(F) = \mathbf{E}_{\sigma_1, \cdots, \sigma_m \in \{+1,-1\}^m}[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x_i)] \tag{2}$$

and

$$R_m[F] = \mathbf{E}_{|S|=m}[R_S(F)] \tag{3}$$

If $H : X \to X$, loss $l : X \to \mathbb{R}$, $F = H \circ l$, then $\forall\ h \in H$,

$$L_S(h) \leq L_D(h) + 2R_S(H) + \mathcal{O}(\sqrt{\frac{\log 1/\delta}{m}}) \tag{4}$$

with probability $1 - \delta$.

Now how does this help us?

Example 2.3. Consider hypothesis class $\mathcal{H} = \{w \to w^T x\}$ of linear separators. Suppose we bound $\|w\|_2 \leq B$ with $\|x_i\|_2 \leq r$ (note that this implies that the distribution is bounded by the sphere). Then we claim that

Theorem 2.4. $R_S(\mathcal{H}) \leq \frac{rB}{\sqrt{m}}$

The main thing that will give us something new is looking at polynomial kernels rather than linear kernels later on.

Proof.

$$\begin{aligned}
mR_S(\mathcal{H}) &= \mathbf{E}_{\sigma_1, \cdots, \sigma_m}[\sup_w \sum \sigma_i w^T x_i] \\
&= \mathbf{E}_{\sigma_1, \cdots, \sigma_m}[\sup_w w^T \left(\sum \sigma_i x_i\right)] \\
&\leq \mathbf{E}_{\sigma_1, \cdots, \sigma_m}[\sup_w \|w\|_2 \cdot \|\sum \sigma_i x_i\|_2][\text{ by Cauchy-Schwarz}] \\
&\leq B \cdot \mathbf{E}_{\sigma_1, \cdots, \sigma_m}[\|\sum \sigma_i x_i\|] \\
&\leq B \cdot \left(\mathbf{E}[\|\sum \sigma_i x_i\|^2]\right)^{1/2} [\text{ by Jensen}] \\
&= B \sqrt{\mathbf{E}_{\sigma_1, \cdots, \sigma_m}[\sum_{i,j} \sigma_i \sigma_j x_i^T x_j]} \\
&= B \sqrt{\sum \|x_i\|^2} \\
&\leq B \sqrt{mr^2}
\end{aligned} \tag{5}$$

Note that line 6 happens □

Note that in this course, we care about real-valued functions, not $0 - 1$ loss as much (in that case it's bounded by VC-dimension which is bounded by dimension of the space). To reiterate, in the case that the numbers become large, the loss can also become large – it makes sense you need more samples. But if you have a bound on the size, than the loss becomes manageable as well.

□

2.1 Rademacher Calculus

This will help us get generalization bounds beyond $0-1$ losses. Let $A \subseteq \mathbb{R}^m$. Think of each $a \in A$ as a vector which has the losses of all x_i: $[l(h(x_1)), l(h(x_2)), \cdots, l(h(x_m))]$. Define

Definition 2.5.

$$R(A) = \mathbf{E}_{\sigma_1,\cdots,\sigma_m}[\sup_{a\in A} \sum \alpha_i a_i] \tag{6}$$

Now consider what happens if you take $R(c_0 \cdot A + c_1)$? If you just add a constant, nothing changes in expectation (± 1 in expectation is 0). However, multiplying might do something: $R(c_0 \cdot A + c_1) = |c_0| R(A)$, assuming that $\sum \sigma_i a_i$ is non-negative, or that for every postive a_j there is a negative a_j (symmetric).

Lemma 2.6. *Talagrand's Lemma.*
Let $l' : \mathbb{R} \to \mathbb{R}$ which is ρ-Lipschitz ($|l'(x) - l'(y)| \leq \rho(x-y)$). This is satisfied if l' is differentiable and its derivative is bounded by ρ. But the Lipschitz property is more general, and doesn't require differentiability, only continuity. Then, let $l'(A) = \{(l'(a_1), l'(a_2), \cdots, l'(a_m)) | a \in A\}$. Consider the example $l'(w^T x) = (w^T x - y)^2$. Then, we have that

$$R(l'(A)) \leq \rho R(A) \tag{7}$$

Note that l' need not be convex! This is a useful property to know to get generalization error bounds.

Now we will prove generalization error bounds for PCA.

What is the hypothesis class $\mathcal{H}$ for PCA? It is the set of all pairs (f, g) indexed by A such that $f_A(x) = Ax$, and $g_A(y) = A^{-1} y$, with $A \in \mathbb{R}^{k\times d}$.

We have $l(h_A, x) = \|x - A^{-1}Ax\|^2 = \|x - g \circ f(x)\|_2^2$, the reconstruction error. Now the question we want to ask is how many examples do we need to see before we can characterize the reconstruction error of this class? We also assume $\|x\|_2 \leq 1$ for simplicity.

Theorem 2.7. $R_S(\mathcal{H}_k^{PCA}) \leq ?$.

Proof.

$$\begin{aligned}
m \cdot R_S(\mathcal{H}_k^{PCA}) &= \mathbf{E}_{\sigma' s}[\sup_A \sum_{i=1}^m \sigma_i \|x_i - A^{-1}Ax_i\|^2] \\
&= \mathbf{E}_\sigma[\sup_A \sum_i \sigma_i x_i^T \left(I - A^{-1}A\right)^2 x_i] \\
&= \mathbf{E}_\sigma[\sup_A \sum_i \sigma_i \mathrm{Tr}(\left(I - A^{-1}A\right)^2 x_i x_i^T)] \\
&\leq \mathbf{E}_\sigma[\sup_A \mathrm{Tr}\left(\left(I - A^{-1}A\right)^2 \sum_i \sigma_i x_i x_i^T\right)]
\end{aligned} \tag{8}$$

Then note that $\langle A, B\rangle = TrAB \leq |A|_{op}|B|_*$ so by Holder's inequality (any primal dual norm pair). Here we take $op =$ Frobenius and $*$ as the Frobenius norm (this is just Cauchy-Schwarz).

$$\begin{aligned}
&\leq \mathbf{E}_\sigma[\sup_A \|I - A^{-1}A\|_F \times \|\sum_i \sigma_i x_i x_i^T\|_F] \\
&\leq \sqrt{d}\mathbf{E}_\sigma[\|\sum \sigma_i x_i x_i^T\|] \\
&\leq \sqrt{d}\mathbf{E}_\sigma[\Big(\sum \sigma_i x_i x_i^T\Big)^2]^{1/2} \\
&= \sqrt{d}\sqrt{\mathbf{E}_\sigma[\sum_{i,j} \sigma_i \sigma_j \langle x_i, x_j\rangle^2]} \\
&= \sqrt{d\sum_i \|x_i\|^2} = \sqrt{md}
\end{aligned} \tag{9}$$

You can optimize the norm pairs correctly to get k instead of d. Now for all A $\|L_S(h_A) - L_D(h_A)\| \leq 2\mathbb{R}_m(\mathcal{H}_k^{PCA}) + \sqrt{\frac{\log 1/\delta}{\epsilon}}$. □

Now why is this interesting? We've proved something stronger than generalization bound; we have agnostic learning. You're competing with best linear encoding for this data (even if your data is not from the subspace).

3 Spectral Autoencoders

In PCA, we have $x \to Ax$. Instead, let's consider a polynomial version of this: $x \to [p_1(x), p_2(x), \cdots, p_k(x)]$. Quadratic would mean take $x^{\otimes 2} = [x_1, \cdots, x_d, x_1x_2, \cdots, x_ix_j, \cdots]$. Then $A \in \mathbb{R}^{k\times(d^2/2+d)}$. Now how do you define an unsupervised learning problem that can learn a manifold of square degree? We can try to learn the best encoding-decoding with minimum reconstruction error.

Let's consider the hypothesis class parametrized by matrices $\mathcal{H} = \{A, f_A(x) = Ax^{\otimes 2}, g(y) = v_{max}(A^{-1}y)\}$. Even more natural would be to pick $g(y) = \text{argmin}_{x\in\mathbb{R}^d}\|Ax^{\otimes 2} - y\|_2^2$. There are many things you can think of, and these give rise to nonconvex optimization problems which are hard to globally optimize. Note that before you take the top eigenvector, you have to do a reshaping operation on $A^{-1}y$ in order to get a matrix. (intuition here is that the matrix form of $x^{\otimes 2}$ is xx^T, which has top eigenvector x. We also have a linear map A in this case which smears things, it turns out you can prove that this approach is robust to noise).

Theorem 3.1. *$\mathcal{H}$ is learnable with sample complexity $\sim \frac{dk}{\epsilon^2}$ using a similar approach.*

The main question is can we come up with efficient algorithms to solve these problems. This is where convex relaxation can help.

Theorem 3.2. *$\mathcal{H}$ is efficiently learnable with sample complexity $\mathcal{O}(\frac{d^2k}{\epsilon^4})$.*

This theorem doesn't contradict the NP-hardness of the problem, since we're allowed to generate a hypothesis which does not come from $\mathcal{H}$.

The roadmap for proving this theorem is (1): Give class $\hat{\mathcal{H}}$ which contains $\mathcal{H}$, which has sample complexity in the form of $\mathcal{O}(d^2k/\epsilon^4)$. Then, (2) efficiently learn $\hat{H}$. Let's define $\mathcal{H} = \{f, g, f(x) = Ax, g(y) = By\}$ where we don't restrict B to be the inverse (relaxing the set). Our loss will be $l(h, x) = \|x^{\otimes 2} - BAx^{\otimes 2}\|^2_{spectral}$. This loss function is a relaxation of our previous loss. It turns out that

$$\|x - v_{max}(A^{-1}Ax^{\otimes 2})\|_2^2 \leq l(h, x) \tag{10}$$

so if we minimize $l(h, x)$, we minimize the other too.

Now to get this efficient thing, we have to prove sample complexity for $\hat{\mathcal{H}}$ – you could do Rademacher, but that's kind of non-intuitive. However in this case, we can get simultaneous generalization and optimization, via regret minimization.

We have $\hat{H}$, a set of hypotheses h, and loss functions $l_x(h) = \|x^{\otimes 2} - BAx^{\otimes 2}\|_2^2$. Now, we are going to do a further relaxation and call $h_{AB} = h_D$, where $D = BA \in \mathbb{R}^{d^2 \times d^2}$. And the way we can relax is say $\|D\|_* \leq k$ (bound the trace norm). We have

- $l(h_0)$ is convex in D.
- $\{D\}$ such that $\text{Tr}(D) \leq k$ is also convex.

Both conditions must hold for doing simultaneous optimization and generalization. We define an algorithm:

1. $0- > D_1$, for $t = 1, 2, 3, \cdots, T$, do:
2. sample $x \sim \mathcal{D}$, denote $l_x(D) = l_t(D)$.
3. update: $D_{t+1} = \Pi_K \left[D_t - \frac{1}{\sqrt{t}} \nabla l_t(D_t)\right]$
4. return $\bar{D} = \frac{1}{T}\sum_{t=1}^T D_t$, the average over all iterations.

where Π is the projection operation with respect to Euclidean norm.

Theorem 3.3. *Online Gradient Descent.*

$$\sum l_t(D_t) - \min_{D^* \in K} \sum l_t(D_t) \leq \frac{2d^2\kappa}{\sqrt{T}} \tag{11}$$

where κ is the rank of the matrix and d is the dimension.

As a corollary,

Theorem 3.4.

$$\boldsymbol{E}[L_G(\bar{D})] \leq \min_{h_G \in \mathcal{H}} L_G(h_G) + \frac{2d^2\kappa}{\sqrt{T}} \tag{12}$$

Proof.

$$\begin{aligned}
\mathbf{E}[L_G(\bar{D})] &= \mathbf{E}_{x\sim G}[l_x(\bar{D})] && (13)\\
&= \mathbf{E}_{x\sim G}[l_x(\frac{1}{T}\sum D_t)] && (14)\\
&\leq \mathbf{E}_x[\frac{1}{T}\sum l_t(D_t)] && (15)\\
&\leq \mathbf{E}_x[l_x(h^*)] + \frac{2d^2\kappa}{\sqrt{T}} && (16)\\
&= \min_{h^*\in\mathcal{H}} L_G(h^*) + \frac{2d^2\kappa}{\sqrt{T}} && (17)
\end{aligned}$$

since loss functions are convex, we can apply Jensen. Line 4 uses the OGD theorem. This holds for any $h*$ since regert is worst-case, which is why we get the minimum h^* over $\mathcal{H}$.

Every time you choose example x, it's independent of the D_t. You choose x from distribution. So you have to write a summation of conditional expectations and then do a martingale analysis, but what we have written is precise because it comes out to the same thing. □

So far, we have discussed one approach to unsupervised learning: compression based, and we've argued why it makes sense, and how you can get generalization error bounds from these definitions which are non-standard for unsupervised learning. We saw how to get these from Rademacher complexity, and discussed how to do it for PCA and spectral autoencoders and generalization by optimization. One of the strong points of this framework is that you can use convex relaxation. This has been done for other classes of unsupervised learning, like dictionary learning.