CS269: Machine Learning Theory
Lecture 3: The Unrealizable Case
October 4, 2010

Lecturer: Jennifer Wortman Vaughan
Scribe: Evan Lloyd, Jacob Mathew

1 General Learning Bound (Continued)

Here are a few things to note about the general learning bound.

- The general learning bound applies only when the hypothesis class is finite. We have looked at some examples of this like the class of monotone conjunctions and the class of DNFs.
- No assumptions are made about the algorithm other than that it returns a hypothesis that is consistent with the samples. In particular, it says nothing about the efficiency of the algorithm. An algorithm that is exponential (in $1/\epsilon$, or $1/\delta$, or n) will still apply to the general learning bound.
- The bound on the number of examples involves a factor of $\ln(|\mathcal{C}|)$. We are better off if we have a $\mathcal{C}$ of smaller size. Bounds with this property are called Occam's razor bounds (the simpler the better). This is an intuitive result. If $\mathcal{C}$ is large, it would seem like there are more hypotheses that just happen to agree with the sample (and hence are consistent), but are not the target concept. So we would need a larger number of examples to rule these out. In contrast if $\mathcal{C}$ is small then if a hypothesis is consistent it is more likely to actually be the target function.

1.1 Applying the General Learning Bound

Since the general learning bound applies when we have algorithms that return consistent hypotheses, we can use this result to investigate the algorithms that we studied under the consistency model. In particular we can try to see what learnability of a class under the consistency model tells us about learnability of the class under the PAC model.

If we have a class that is learnable under the consistency model this means that there is an algorithm for that class that returns a consistent hypothesis if one exists. We are dealing with the case that the target function is a member of the hypothesis class, so we are guaranteed that there will always be at least one consistent hypothesis. We can now immediately apply the general learning bound to bound the number of samples needed to get $\Pr(\mathrm{err}(h) > \epsilon) < \delta$.

1.1.1 The Class of Monotone Conjunctions

Consider the class of monotone conjunctions. We showed earlier that this is learnable under the consistency model. For this hypothesis class we have $|\mathcal{C}| = 2^n$. This gives us that

$$m \geq (1/\epsilon) \cdot (n + \ln[1/\delta]).$$

This means that a polynomial number of samples will do to get a good enough hypothesis with high enough probability. Therefore the class of monotone conjunctions is PAC learnable.

1.1.2 The Class of DNFs

Consider the class of DNFs. Here we have $|\mathcal{C}| = 2^{2^n}$, so the bound we get is

$$m \geq (1/\epsilon) \cdot (2^n + \ln[1/\delta])$$

This does not tell us that the class is PAC learnable. Note that it also does not tell us that the class is *not* PAC learnable, since the bound is not a tight bound. It only shows that we cannot use the general learning bound to show PAC learnability. The question of whether this class is PAC learnable or not is an open problem.[1]

1.2 PAC Learning and Consistency Learning

We can also relate the consistency model to the PAC model in another way. Specifically, we can show that a slightly altered version of consistency learning reduces to PAC learning.

Assume we have an algorithm $\mathcal{A}$ that PAC-learns $\mathcal{C}$ by $\mathcal{C}$. Define a new algorithm $\mathcal{A}'$, which, given a set S of m labeled examples, operates as follows:

1. Set ϵ to some value smaller than $1/m$
2. Define a distribution $\mathcal{D}$ which is uniform over S
3. Run $\mathcal{A}$, giving it $(1/\epsilon) \cdot (\ln[|C|] + \ln[1/\delta])$ samples drawn from $\mathcal{D}$
4. If $\mathcal{A}$ returns an hypothesis $h \in \mathcal{C}$ such that $err(h) < 1/m$, return h. Otherwise, return "none."

Note that we give $\mathcal{A}$ a number of samples determined by the general learning bound, which means that $\mathcal{A}$ will return a hypothesis such that $err(h) < \epsilon < 1/m$ with at least probability $1 - \delta$. Because we have a finite number of samples and we defined $\mathcal{D}$ to be uniform, it must be the case that for every $h \in \mathcal{C}$, $err(h) \in \{0/m, 1/m, 2/m, \ldots m/m\}$, depending on how many of the samples in S it misclassifies. Therefore, if $err(h) < 1/m$, the error must actually be *zero,* or equivalently, h must be consistent with S. This means that $\mathcal{A}'$ learns a consistent hypothesis with probability at least $1 - \delta$. This is very similar to learning $\mathcal{C}$ in the consistency model, except that we now have a probability of failure δ.

2 The Unrealizable Case

So far, we have been working with the assumption that we have a perfect target function. That is, we assume that the data is being labeled consistently by a specific function in the concept class. This is not always a valid assumption to make. It may be the case that there is indeed a perfect target function, but it is not in the

[1] Some variations of the PAC model allow the algorithm to run in time $\text{poly}(1/\epsilon, 1/\delta, n,$ **#bits required to store target function c**). You may have come across one of these definitions in your reading. For DNFs, this would mean that we might allow the algorithm time polynomial in the number of terms in the minimal DNF representation of the target c. For some particular functions, the number of terms in a minimal DNF could be as many as 2^n, in which case we would allow the algorithm time exponential in the size of an example. However, for many simpler target functions, the size of the representation is much smaller and we would not allow this exponential dependence. Since the algorithm must run in polynomial time in these parameters for *every* $c \in \mathcal{C}$, the Occam's razor bound does not imply PAC learnability of DNFs in these variants of the PAC model either. (See Chapter 7.1 of Kearns and Vazirani for some discussion of this.)

concept class that we are considering. Or there might be noise in the data. The data might get mislabeled occasionally, giving rise to some randomness. In view of this, we would like to loosen this assumption and see how that affects the results we have obtained so far. This is often referred to as the *unrealizable* setting, whereas the setting in which there is a perfect target function is referred to as the *realizable* setting.

By dropping the assumption of perfect target functions, we also need to update our assumptions about how data is provided to the algorithm. Recall that earlier we assumed that the data was generated from a distribution $\mathcal{D}$ and was labeled by the target function. We model this now as a joint distribution over pairs of values $(\vec{x}, y)$ where $\vec{x}$ is the data and y is the labeling of $\vec{x}$, so we no longer refer to a target function labeling the data.

This gives us a new definition of the error of a hypothesis. Recall that earlier we had

$$\mathrm{err}(h) = \mathrm{Pr}_{\vec{x}\sim\mathcal{D}}\left[h(\vec{x}) \neq c(\vec{x})\right].$$

The new definition of error is

$$\mathrm{err}(h) = \mathrm{Pr}_{(\vec{x},y)\sim\mathcal{D}}\left[h(\vec{x}) \neq y\right].$$

Note that this is a strictly more general way to define the error. We can still model a perfect target function as a joint probability distribution for which the label y is deterministic conditioned on the data $\vec{x}$ and corresponds to the target function's labeling of $\vec{x}$.

In the results that we had before we talked about the probability of finding a hypothesis h such that $\mathrm{err}(h) < \epsilon$. We are no longer guaranteed that such a function even exists, so we need to relax this. The corresponding notion that we will use is how close the error of the hypothesis is to that of the hypothesis that has the least error in the hypothesis class.

Our earlier results also talked about the idea of a consistent hypothesis. It is no longer the case that there is necessarily any consistent hypothesis in the hypothesis class, so we relax this notion as well by considering instead the hypothesis that is best or most consistent with the sample data. We measure the extent of the consistency of a hypothesis by its *empirical error*, which is defined as

$$\widehat{\mathrm{err}}(h) = (1/m) \cdot |\{i : h(\vec{x_i}) \neq y_i\}|$$

This value converges to the *true error* $err(h)$:

$$E[\widehat{\mathrm{err}}(h)] = err(h).$$

We denote the hypothesis with least empirical error as $\widehat{h}$:

$$\widehat{h} = \underset{h\in\mathcal{H}}{\mathrm{argmin}}\left(\widehat{\mathrm{err}}[h]\right).$$

2.1 General Learning Bound in the Unrealizable Setting

In this section we are going to derive a general learning bound without assuming perfect target functions. Assume that we have an algorithm $\mathcal{A}$ that, given m independent samples from an arbitrary distribution $\mathcal{D}$, outputs the hypothesis $\hat{h} \in \mathcal{H}$ with minimal empirical error. We will be looking for a lower bound on m that guarantees that for any $\epsilon, \delta \in (0,1)$, $\mathrm{Pr}[|\mathrm{err}(\widehat{h}) - \min_{h\in\mathcal{H}}(\mathrm{err}\{h\}) \mid > \epsilon] < \delta$.

Assume the following property holds of the hypothesis class:

$$\forall h \in \mathcal{H},\ |\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \leq \epsilon.$$

We can then derive the following:
For any $h \in \mathcal{H}$,

$$\begin{aligned} \mathrm{err}(\widehat{h}) &\leq \widehat{\mathrm{err}}(\widehat{h}) + \epsilon && \text{this follows from the assumption} \\ &\leq \widehat{\mathrm{err}}(h) + \epsilon && \text{this follows from the definition of } \widehat{h} \\ &\leq \mathrm{err}(h) + 2\epsilon. && \text{this follows from the assumption} \end{aligned}$$

Since the derivation above holds for any $h \in \mathcal{H}$, it must hold for the h that minimizes error, so we have that

$$\mathrm{err}(\widehat{h}) \leq \min_{h \in \mathcal{H}}(\mathrm{err}[h]) + 2\epsilon.$$

This tells us that if we can bound the number of examples so that our assumption holds with high probability we are effectively done. Our first attempt at trying to achieve this bound is going to use a basic result from probability theory called Markov's inequality. It will turn out to not be sufficient for our needs but will give us an indication of how to proceed.

Theorem 1. *(Markov's inequality) If X is a non-negative random variable and k is any positive real number, then*

$$Pr\left[X \geq k\right] \leq E\left[X\right]/k.$$

Proof:

$$\begin{aligned} E\left[X\right] &= \Pr\left[X \geq k\right] \cdot E\left[X | x \geq k\right] + \Pr\left[X < k\right] \cdot E\left[X | x < k\right] \\ &\geq \Pr\left[X \geq k\right] \cdot E\left[X | x \geq k\right] + 0 \\ &\geq \Pr\left[X \geq k\right] \cdot k, \end{aligned}$$

which we can rearrange to get

$$\Pr\left[X \geq k\right] \leq E\left[X\right]/k.$$

□

We can apply Markov's inequality as follows. For a fixed h, let $X = \widehat{\mathrm{err}}(h)$, $k = err(h) + \epsilon$. Then substituting into Theorem 1, we get

$$\Pr[\widehat{err}(h) \geq err(h) + \epsilon] \leq \frac{E[\widehat{err}(h)]}{err(h) + \epsilon} = \frac{err(h)}{err(h) + \epsilon}.$$

This does not give us quite what we want because in general we do not have a bound on $err(h)$, and so for large enough errors the bound can get close to 1. Also the fact that this does not depend on the number of samples is indicative that we are not doing everything quite right here.

We are going to improve upon this by using a related inequality called Hoeffding's inequality. We will derive Hoeffding's inequality from Markov's inequality in the next lecture.

Theorem 2. *(Hoeffding's inequality) Let $Z_1, Z_2, \ldots, Z_m$ be independent random variables such that $Z_i \in [a_i, b_i]$, $i = 1 \ldots m$. Define an empirical mean $\hat{p} \equiv (1/m) \cdot \sum_{i=1}^{m} Z_i$ and its expectation $p \equiv E[\hat{p}] = (1/m) \cdot \sum_{i=1}^{m} E[Z_i]$. For any $\epsilon > 0$,*

$$\Pr[p - \hat{p} \geq \epsilon] \leq e^{-2\epsilon^2 m^2 / \sum_{i=1}^{m}(b_i - a_i)^2}$$

and

$$\Pr[\hat{p} - p \geq \epsilon] \leq e^{-2\epsilon^2 m^2 / \sum_{i=1}^{m}(b_i - a_i)^2}.$$

Corollary 1. *Under the same assumptions on Z_1, Z_2, ..., Z_m, p, and $\hat{p}$ as above,*

$$Pr\left[|p - \widehat{p}| \geq \epsilon\right] \leq 2e^{-2\epsilon^2 m^2 / \sum_{i=1}^{m}(b_i - a_i)^2}.$$

Proof: This follows immediately from the application of union bound to the two probabilities in Hoeffding's inequality. □

We are now ready to show the bound we are looking for on the number of samples.

Theorem 3. *(General learning bound in the unrealizable setting) If $m \geq (1/2\epsilon^2) \cdot (\ln[|\mathcal{H}|] + \ln[2/\delta])$, then with probability at least $1 - \delta$,*

$$\forall h \in \mathcal{H}, |\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \leq \epsilon.$$

Proof: Consider some $h \in \mathcal{H}$. Define $Z_i = \begin{cases} 1 & \text{if } h(x_i) \neq y_i \\ 0 & \text{otherwise} \end{cases}$, $i = 1 \ldots m$.

Since $\widehat{err}(h) = (1/m) \cdot \sum_{i=1}^{m} Z_i$ and $err(h) = E[\widehat{err}(h)]$, we can apply (the corollary to) Hoeffding's inequality with $\hat{p} = \widehat{err}(h)$, $p = err(h)$, $Z_1 \ldots Z_m$, and $a_i = 0$, $b_i = 1$, $i = 1 \ldots m$. This gives

$$\Pr[|\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \geq \epsilon] \leq 2e^{-2\epsilon^2 m}.$$

Applying this over all $h \in \mathcal{H}$ using the union bound, we get

$$\Pr[\exists h \in \mathcal{H} \,:\, |\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \geq \epsilon] \leq 2|\mathcal{H}|e^{-2\epsilon^2 m}.$$

Since we want to show $|\mathrm{err}(h) - \widehat{\mathrm{err}}(h)| \leq \epsilon$, this is the probability of *failure.* If we let $2|\mathcal{H}|e^{-2\epsilon^2 m} \leq \delta$, by re-arranging and taking the log, we get

$$m \geq (1/2\epsilon^2) \cdot (\ln[|\mathcal{H}|] + \ln[2/\delta]).$$

□

2.2 Remarks about the General Learning Bound

When we compare the bound on the number of samples for the unrealizable case with the bound for the realizable case, we see that they have a very similar form. The Occam's razor principle applies here as well. The fewer hypotheses we have the better because of the $\ln(|\mathcal{H}|)$ factor. Note that also we now have ϵ^2 where we used to have ϵ. This means that we are paying a penalty of a factor $1/\epsilon$ by not having a perfect target function.

By solving for ϵ in the bound we obtained for m and applying it to the inequality we obtained in an earlier result, we get

$$\mathrm{err}(\widehat{h}) \leq \min_{h \in \mathcal{H}} (\mathrm{err}[h]) + O\left(\sqrt{\frac{\ln(|\mathcal{H}|) + \ln(2/\delta)}{m}}\right).$$

This gives us an interesting perspective on the effect of the size of the hypothesis class. The first term in this equation tells us that we want a large hypothesis class because we increase our likelihood of minimizing the error. However the second term tells us that a smaller hypothesis class is better because of the Occam's razor principle.

Note that way in which the Occam's razor principle shows up here can be described in terms of *overfitting*. When we have a large $\mathcal{H}$ and we try to fit our hypothesis to the data, the likelihood of picking the wrong hypothesis is larger, so overfitting becomes a bigger issue.