

## PAC Learning: Rectangles

## 1 Algorithm

This note is to try to explain the axis-aligned rectangle derivation in a bit more detail than is on the slides. It also attempts some faux-proofs that *don't* work and tries to explain why. (Caveat: it's often hard to explain why you can't prove something some way – you just have to find out that the proof doesn't go through.)

Here's our goal:

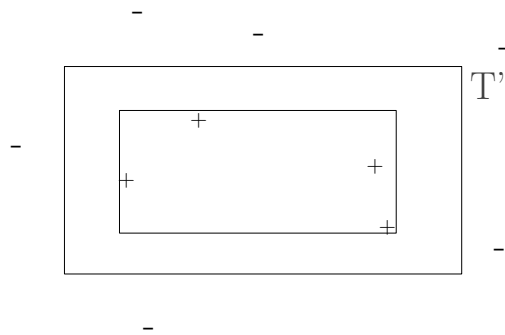
- We are given an error parameter  $0 < \epsilon < \frac{1}{2}$
- We are given a confidence parameter  $0 < \delta < \frac{1}{2}$
- Access to as many labeled points as we want from  $\mathcal{D}$

We have to show that we can learn a *hypothesis* axis aligned rectangle ( $h$ ), assuming that the *true concept* is an axis aligned rectangle  $c$ , is time polynomial in  $\epsilon^{-1}$  and  $\delta^{-1}$ . In particular, we have to use a number of samples that is polynomial in these terms. Our hypothesis must have error at most  $\epsilon$  with probability at least  $1 - \delta$ . In symbols, this means:

$$Pr(\underbrace{\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x)]}_{\text{expected loss}} > \epsilon) < \delta \quad (1)$$

What this says is that we draw a bunch of random points  $x$  from the distribution  $\mathcal{D}$ . We measure our expected (average) loss on these points. We check to see if this expected loss is greater than  $\epsilon$ . There's some probability that it will be greater than  $\epsilon$  and we need this probability to be at most  $\delta$ . The  $Pr(\cdot)$  probability is a probability over: (a) the random draws of the training points and (b) any internal randomization in our algorithm (in this case, we have none). So it's basically trying to guard against the chance that we are unlucky and get a useless training set.

This is basically just PAC learning. Now we move on to our algorithm. Recall that what we're doing is maintaining a single axis aligned rectangle  $h$  that “just covers” the positive points. For instance, see the Figure below:



Here, we have a true concept rectangle  $c$  (the outer rectangle) and a hypothesis rectangle  $h$  (the inner rectangle).  $h$  is configured so that it is the (unique) smallest rectangle that covers all the positive points.

## 2 Good Analysis

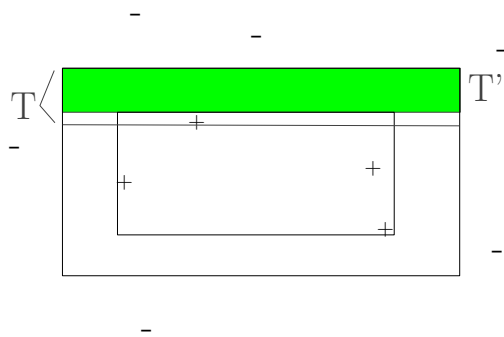
Our goal is to show that after  $N$  points, where  $N$  is polynomial in  $\epsilon^{-1}$  and  $\delta^{-1}$ , Eq (1) holds, where  $h$  is according to our “smallest rectangle” algorithm. First, we’re going to prove this in the same way we proved it in class. Then, we’ll move on and show that some “more obvious” proofs don’t actually work.

In order to prove Eq (1), we need to evaluate what the expected loss is going to be for some hypothesis  $h$ . We first note that the only place where our algorithm makes an error is when it is “too small” – namely, a test point comes in the area  $c - h$  that should be labeled positive. So the “bad areas” are the strips on the top, bottom, left and right. I claim (we’ll see why later) that this is hard to work with all of these simultaneously, so we’ll start by looking just at the top.

Our goal is to show that the error we incur in the top is at most  $\epsilon/4$ , the error we incur on the bottom is at most  $\epsilon/4$ , same for left and same for right. We then apply the *union bound*. This says, essentially, that if I have two events  $A$  and  $B$ , then  $Pr(A \text{ or } B) \leq Pr(A) + Pr(B)$  (equality only holds in the case of disjointness of  $A$  and  $B$ ). Thinking about this pictorially, we basically know that if  $A$  and  $B$  overlap, then  $Pr(A \text{ or } B)$  single-counts where they overlap, but  $Pr(A) + Pr(B)$  double-counts this area. In symbols, we know that  $Pr(A \text{ or } B) = Pr(A) + Pr(B) - Pr(A \text{ and } B)$ , from which the bound clearly follows. In our case, the overlap corresponds to the corners of  $c - h$ . What we’re saying is that we’re *okay* double-counting these areas. We’d rather not: we get a looser bounds. But it will still be acceptable. (We’ll work through this in symbols later.)

Define  $T'$  to be the “bad area” on the top – namely, the strip along the top of  $h$  that is still inside  $c$ . My goal is to show that this strip is small. In particular, if I’m forced to have at most  $\epsilon$  error overall, I’m going to allow myself  $\epsilon/4$  error on the top. What does it mean to have  $\epsilon/4$  error on the top? It means that the probability of drawing a point  $x \sim \mathcal{D}$  that lies in  $T'$  is at most  $\epsilon/4$ .

Consider the (unique) top strip of  $c$  (this definition has nothing to do with  $h$ ) that contains  $\epsilon/4$  of the probability mass of  $\mathcal{D}$  for positive points:



In this figure,  $T'$  (the error bar) is shaded green and is drawn smaller than  $T$ . Note that we don’t *know* that  $T'$  is smaller than  $T$ : this is actually exactly what we want to prove. If we know that  $T'$  is smaller than  $T$ , then we know that the probability of  $T'$  is at most  $\epsilon/4$  (since that’s the probability of  $T$ ). So our *goal* is to show that  $T$  contains  $T'$ .

So we want to show that  $T$  contains  $T'$ . Let’s assume it’s not true. This means that  $T'$  contains  $T$ . In other words, the error area is *too big*. According to our algorithm, if  $T'$  contains  $T$  then that means that our training data never included any samples from  $T$ . If it had,  $h$  would have stretched up enough to cover part of  $T$ . This means that *every* training point missed  $T$ .

The probability that a single training point falls in  $T$  is  $\epsilon/4$  by definition of  $T$ . This means that the probability that a single point *misses*  $T$  is  $1 - \epsilon/4$ . But we’re supposing that we get  $N$  training points: the probability that the *all* miss  $T$  is  $(1 - \epsilon/4)^N$ .

What we have at this point is that the probability that  $N$  training points *don’t fall* in the top  $\epsilon/4$  region is

exactly  $(1 - \epsilon/4)^N$ . In symbols:

$$\Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{top bar}] > \epsilon/4) < (1 - \epsilon/4)^N \quad (2)$$

In other words, we look at the expected error in the top portion *only*. The probability that this is greater than  $\epsilon/4$  is at most  $(1 - \epsilon/4)^N$ .

Now, we apply the same methodology to the bottom, left and right. This gives us a total of four statements of the form Eq (2). We can apply the union bound to these probabilities. This yields:

$$\begin{aligned} \Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x)] > \epsilon) &= \Pr(\mathbb{E}_{x \sim \mathcal{D}}[(h(x) \neq c(x) \wedge x \in \text{top bar}) \text{ or} \\ &\quad (h(x) \neq c(x) \wedge x \in \text{bottom bar}) \text{ or} \\ &\quad (h(x) \neq c(x) \wedge x \in \text{left bar}) \text{ or} \\ &\quad (h(x) \neq c(x) \wedge x \in \text{right bar})] > \epsilon) \end{aligned} \quad (3)$$

$$\begin{aligned} &\leq \Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{top bar}] + \\ &\quad \mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{bottom bar}] + \\ &\quad \mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{left bar}] + \\ &\quad \mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{right bar}] > \epsilon) \end{aligned} \quad (4)$$

$$\begin{aligned} &= \Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{top bar}] > \epsilon/4) + \\ &\quad \Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{bottom bar}] > \epsilon/4) + \\ &\quad \Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{left bar}] > \epsilon/4) + \\ &\quad \Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x) \wedge x \in \text{right bar}] > \epsilon/4) \end{aligned} \quad (5)$$

$$\leq 4(1 - \epsilon/4)^N \quad (6)$$

In Eq (3), we're simply expanding out the definition of error. We're using the fact that  $h \subseteq c$ . That is, we know that the error must happen in one of these bars. In Eq (4), we apply the union bound. In Eq (5), we break the single probability into the sum of four probabilities and divide each bound by four. In Eq (6), we use the fact that the probability of each of this is at most  $(1 - \epsilon/4)^N$ , multiplying this by four, since there are four terms.

Now, compare this result to our original goal, Eq (1), which is reproduced below:

$$\underbrace{\Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x)] > \epsilon)}_{\text{expected loss}} < \delta$$

What we have is that:

$$\underbrace{\Pr(\mathbb{E}_{x \sim \mathcal{D}}[h(x) \neq c(x)] > \epsilon)}_{\text{expected loss}} \leq 4(1 - \epsilon/4)^N$$

So what we need to do is choose  $N$  big enough that  $4(1 - \epsilon/4)^N < \delta$  and we're done. The problem is that this is a bit tough to solve for  $N$ , so we apply the bound:  $(1 - x) \leq e^{-x}$ . Here, we identify  $x$  as  $(\epsilon/4)$  and get:

$$4(1 - \epsilon/4)^N \leq 4 \exp[-\epsilon/4]^N = 4 \exp[-\epsilon N/4]$$

We take logs and solve for  $N$ , yielding:

$$N \geq (4/\epsilon) \log(4/\delta)$$

This shows that so long as we draw at least  $(4/\epsilon) \log(4/\delta)$  training points (for a given  $\epsilon$  and  $\delta$ ), we are guaranteed to obey the PAC requirement. And we're done.

### 3 Bad Analysis

The first question that arises upon seeing this analysis is: why do we have to analyze the four strips separately? Why can't I just analyze the total bad region at once?

Let's suppose we tried to do this. Let  $B'$  be the *bad region* (analogous to  $T'$  above) and let  $B$  be the  $\epsilon$  boundary of  $c$  (analogous to  $T$  above). By definition of  $B$ , the probability that a random training point misses  $B$  is  $(1 - \epsilon)$  so the probability that  $N$  training points miss  $B$  is  $(1 - \epsilon)^N$ . We then bound  $N$  so that this probability is  $< \delta$  and obtain something that's tighter than the "good analysis."

Why doesn't this work?

It's a subtle error, namely the use of the word "the" in the phrase "let  $B$  be *the*  $\epsilon$  boundary of  $c$ ". The probable with this is that there is no single boundary of  $c$  with probability  $\epsilon$ . There are a lot of them! If we have one, we can remove a little from the top and add a little to the left, for instance.

So why does this lead to an incorrect analysis? Well, what we've shown is that *for some choice* of  $B$  (where  $B$  has probability  $\epsilon$ ), the probability of error is at most  $(1 - \epsilon)^N$ . But maybe we just got lucky in our choice of  $B$ . We'd need to show that for *all choices* of  $B$  (where  $B$  has probability  $\epsilon$ ) we get a useful bound. The reason we would need to show this is because just missing *one*  $\epsilon$  set isn't enough—we have to miss them all; otherwise we'll still have error  $> \epsilon$ .

This is why we started looking at strips in the first place: *the* top  $\epsilon/4$  strip *is* well defined. This gives us something very concrete to work with.

A second question is: does the fact that these are rectangles matter? What if I just take arbitrary shapes? First, we'd have to define our algorithm in some meaningful way for arbitrary shapes. The most natural is to say that we take the convex hull of all the positive points. But if the concept isn't itself convex, when we can make errors both on the positive side and negative side (sketch a picture if you don't believe me). So let's assume that  $c$  is convex too. The problem that we'll run in to is that just getting a single training point is not enough to cut off a large amount of error. In the rectangle case, things worked out nicely because getting a training point *anywhere* in  $T$  was enough to shift the entire top of the rectangle up. This is no longer true if we're just looking at convex hulls. (If you don't believe me, try to prove that learning convex shapes by convex hulls is PAC learnable and I promise it will break somewhere.)