Machine Learning A - Home Assignment 4

Jakob H. Schauser, pwn274

October 2021

1 Occam's Razor

1.1

I look at the problem and work out, that the number of hypotheses must be $2^{(27^d)}$ where d is the length of the string. Here, using Occam's Razor, this gives a $\pi(h) = 1/2^{(27^d)}$ as summing over every hypothesis for a single, chosen dimensionality gives 1:

$$\sum_{h \in H_d} \pi(h) = \sum_{h=0}^{2^{(27^d)}} 1/2^{(27^d)} = 1 \tag{1}$$

As this is withing the constraints for $\pi(h)$, I Occam's Razor gives me:

$$\mathbb{P}\left(\exists h \in H : L(h) \geq \hat{L} + \sqrt{\frac{\ln(1/\pi(h)\delta)}{2n}}\right) =$$
$$\mathbb{P}\left(\exists h \in H : L(h) \geq \hat{L} + \sqrt{\frac{27^d \ln(2) - \ln(\delta)}{2n}}\right) \leq \delta$$

1.2

Now, for having this work for every dimension, I need $\sum_h \pi(h) \leq 1$ for all d. To accomplish this, I follow what Yevgeny did in the lecture, adding a second term. This term is based on the geometric series as this is known to sum to 1, so my final $\pi(h)$ is given as:

$$\pi(h) = \frac{1}{2^{(27^d)}} \frac{1}{2^{d+1}} \tag{2}$$

The $d+1$-term is needed, as even at a length of 0, there are two hypotheses; a guess on English and a guess on Danish. If we didn't correct for this, the term would sum to 2. Now, showing that $\pi(h)$ sums to one is easy, as we can split it up into its two parts, the first of which we already showed, the second part is simply the geometric series:

$$\sum_h \pi(h) = \sum_{d=0}^{\infty} \sum_{h \in H_d} \frac{1}{2^{(27^d)}} \frac{1}{2^{d+1}} = \sum_{d=0}^{\infty} \frac{1}{2^{d+1}} \sum_{h \in H_d} \frac{1}{2^{(27^d)}} = \sum_{d=0}^{\infty} \frac{1}{2^{d+1}} \cdot 1 = 1 \tag{3}$$

This gives a final bound of:

$$\mathbb{P}\left(\exists h \in H : L(h) \geq \hat{L} + \sqrt{\frac{\ln\left(\frac{2^{(27^d)} 2^{d+1}}{\delta}\right)}{2n}} \right) =$$

$$\mathbb{P}\left(\exists h \in H : L(h) \geq \hat{L} + \sqrt{\frac{(27^d + d + 1)\ln(2) - \ln(\delta)}{2n}} \right) \leq \delta$$

fix me pls, I am ugly :(

1.3

It is clear to see, that as the dimensionality rises, the 27^d-term explodes. But this improved complexity might help the empiric loss $\hat{L}$ to fall. As every previous case is integrated into the higher dimensions, the optimal loss will never rise - at worst stagnate i.e. $\hat{L}_d \leq \hat{L}_{d+1}$.

1.4

The 1/8 bound is only applicable at a 1/8 probability, meaning that while the loss might break either of the bounds, there will still exist a hypothesis which will hold both.

2 Early Stopping

2.1

a)

Yes! Stopping after a predefined number of steps will be completely independent of the validation set (as long as you are not training on it, of course).

b)

Here, even though it seems to be a more optimal way of training, minimizing the loss with regards to the validation set makes the estimator biased.

c)

Again, if the early stopping is based on the validation set instead of the training set, it will become biased with regard to this data as well.

2.2

The first bound is simply using Occam's Razor, as we know we have exactly 100 potential hypotheses, meaning we can define $\pi(h) = 1/100$:

$$\mathbb{P}\left(\exists h \in H : L \geq \hat{L}(h) + \sqrt{\frac{\ln(100/\delta)}{2n}}\right) \leq \delta \tag{4}$$

For the second bound we define $\pi(h) = 1/T$ for finding the optimal hypothesis:

$$\mathbb{P}\left(\exists h \in H : L \geq \hat{L}(h^*) + \sqrt{\frac{\ln(T/\delta)}{2n}}\right) \leq \delta \tag{5}$$

For the final one, our hypothesis space is infinite. Using the hint and defining $\pi(h) = 1/t(t+1)$, the bound becomes:

$$\mathbb{P}\left(\exists h \in H : L \geq \hat{L}(h_{t^*}) + \sqrt{\frac{\ln(t(t+1)) - \ln(\delta)}{2n}}\right) \leq \delta \tag{6}$$

These can of course all be seen as a bound on the empirical loss with a confidence of $1 - \delta$

2.3

The only way I can get an answer that makes sense is if the loss is bounded. If it is bounded, we can choose to bound by 1, and then it makes no sense to train after the bound reaches 1. This means we can find t by:

$$\sqrt{\frac{\ln(t(t+1)) - \ln(\delta)}{2n}} = 1 \leftrightarrow t(t+1) = \delta e^{2n} \leftrightarrow t = \sqrt{1 + 4\delta e^{2n}}/2 - \frac{1}{2} \tag{7}$$

2.4

If we have $\pi(h) = 1/2^t$ instead, this calculation gives:

$$\sqrt{\frac{t\ln(2) - \ln(\delta)}{2n}} = 1 \leftrightarrow t = (2n + \ln(\delta))/\ln(2) \tag{8}$$

This one grows as n where the last grew as e^n so this is much slower.

2.5

I did not have time this week to answer this question.

a)

b)

c)

3 Random Forests

3.1 Analyzing Satellite Data

3.1.1

I used the sklearn.ensemble's RandomForestClassifier. I get a validation accuracy of about 0.744.

3.1.2

The produced image can be seen in figure 1. Once I had changed the color palette, I could see that this was an aerial image of the area around Roskilde.

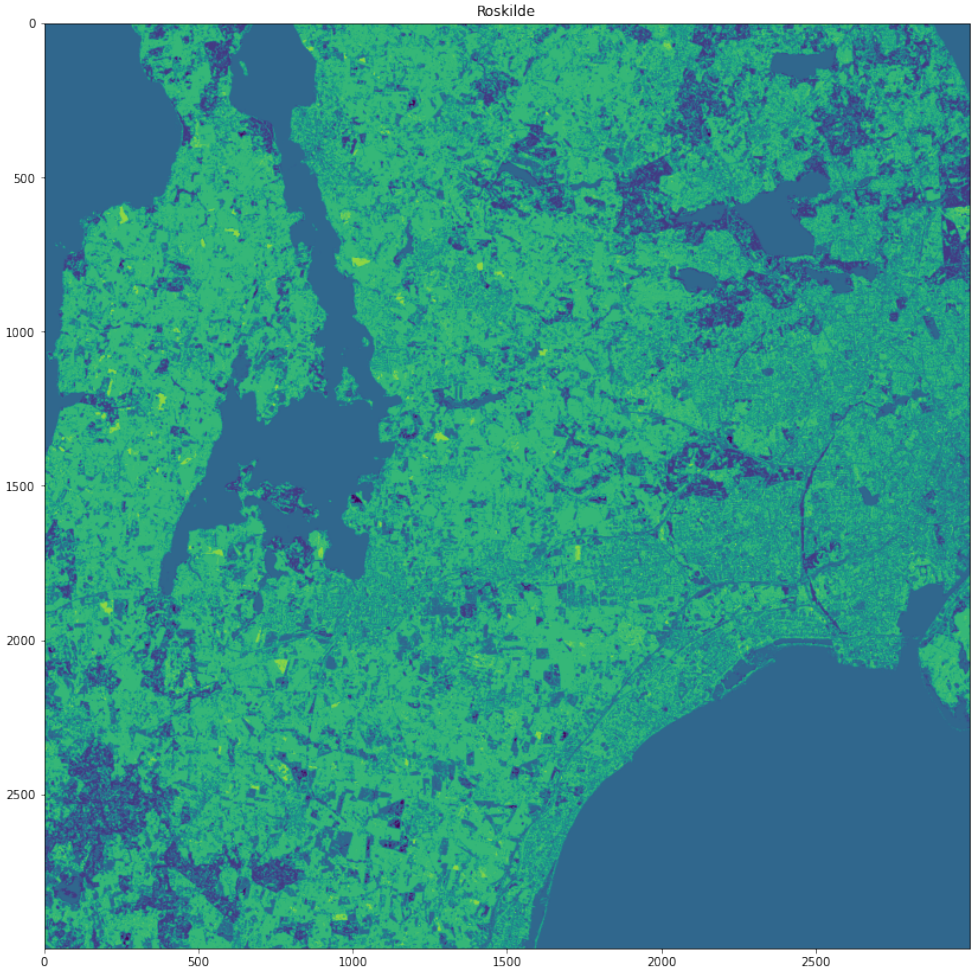

Figure 1: My Random Forest can take pictures of Roskilde

3.1.3

Given that there are only two classes, the worst realistic that can happen at any leaf is a 50-50 split, meaning that for n data points, the maximal number of needed leafs is $\lceil \sqrt{n} \rceil$ where $\lceil x \rceil$ is the ceiling function. This corresponds to finding the largest power of two that is higher than you number. If realism is of no matter, the answer is $n-1$ as the tree at every split can simply choose to classify a single data point and the last one splitting into two.

I am unsure in what terms the asymptotic runtime would work in this case, but my gut feeling says it will be $\mathcal{O}(10n)$.

3.2 Normalization

In both cases, the answer is no.

3.2.1 Nearest Neighbor Classification

It is obvious that any parallel displacements does not change the neighborhoods. Scalings of the parameters would also not change anything, as all distances would be scaled by the same amount, thereby not changing the order of proximity for the data points. (Here I am assuming a relatively normal distance-metric, like the Euclidean, but for some, this might not be the case).

3.2.2 Random Fores Classification

Since random forest classifications simply choose a cut in a parameter, any scalings to the parameters could also be applied to the cut, making the cut still work. The same applies to translations. But any disturbances that 'rotate' parameters into each other or change the order of data points within a single parameters cannot be used.