Assignment 5, Machine Learning Fall 2018

Cristian Mitroi, dmn470

4 January 2019

Contents

1 Train-Validation Split Trade-off

1.1

Since we only have one hypothesis, we can use **Theorem 3.1** from the lecture notes:

$$P(L(h) \leq \hat{L}(h, S) + \sqrt{\frac{ln\frac{1}{\delta}}{2n}}) \geq 1 - \delta$$

Replacing for our case we get the following:

$$P(L(h^*_{S^{train}}) \leq \hat{L}(h^*_{S^{train}}, S^{val}) + \sqrt{\frac{ln\frac{1}{\delta}}{2n^{val}}}) \geq 1 - \delta$$

1.2

Since each hypothesis is trained and validated on its own sample set we cannot use **Theorem 3.2**, which requires that the hypotheses are all validated on the same set. Instead we can use an union bound (as

shown in the *Learning from Data* book, page 24 (Abu-Mostafa, Magdon-Ismail, and Lin 2012)) on **T 3.1** , like so:

$$P(\hat{L}(h_i^*) - L(h_i^*) > \epsilon) \leq \sum_{i=1}^{m} P(\hat{L}(h_i) - L(h_i) > \epsilon) = 2Me^{-2\epsilon^2 n_i}$$

We replace ϵ with $\sqrt{\frac{ln\frac{1}{\delta}}{2n^i}}$:

$$P(\hat{L}(h_i^*) - L(h_i^*) > \sqrt{\frac{ln\frac{1}{\delta}}{2n^i}}) \leq 2Me^{-2\sqrt{\frac{ln\frac{1}{\delta}}{2n^i}}^2 n_i}$$

1.3

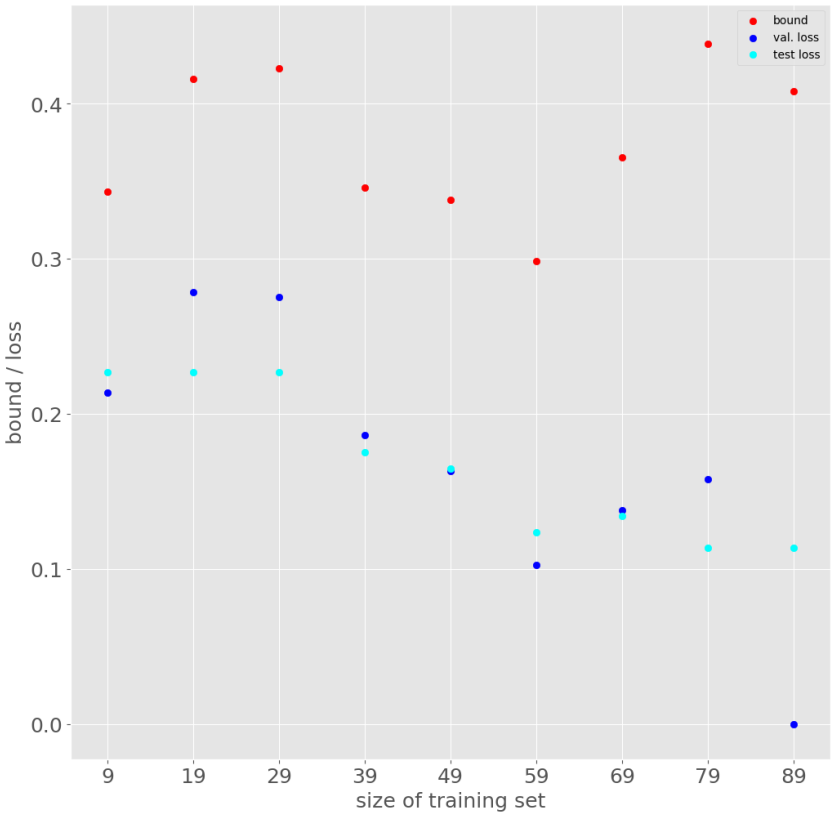

Figure 1: ex 1.3

We can see that the best results are at *size of training set* = 59 and 69. At these points the loss is minimal and the val. loss best estimates the expected test loss.

The biggest difference between val loss and test loss is when the training set is 89. We also notice that the bound in this case is very loose. We can also see that in this case the bound is very loose. This is

because the validation set is so small.

2 Contradiction between Upper and Lower Bound

2.1.

This is incorrect. The VC dimension is defined for all hypothesis classes. The contradiction is valid, since Occam's razor depends on a distribution of the confidence budget.

2.2.

It is true that the bound in Occam's razor could be larger than 0.125 for some hypotheses. This depends on how much mass out of our confidence budget $\pi(h)$ we assign to each of the hypothesis classes. This implies that Occam's razor and the VC lower bound are in contradiction depending on the distribution of the confidence interval budget.

2.3.

This is true. The distribution in constructing the lower bound was selected so that we have $2n$ points that can be shattered by H. This implies that Occam's razor and the VC lower bound are in contradiction depending on how the samples are selected.

2.4.

This can be done by replacing all the values in the VC bound and observing that the bound is not actually 0.01.

We have $m_H(2n) = 2^{2n}$ since $d_{VC}(h) = \infty$. We have $\delta = 0.05$, since the RHS should be 0.95 and the original RHS in the VC bound is $1 - \delta$.

We then get the bound between $L(h)$ and $\hat{L}(h)$ as:

$$\sqrt{\frac{8}{n} ln \frac{4m_H(2n)}{\delta}} = \sqrt{\frac{8}{101} ln \frac{42^{202}}{0.05}} \approxeq 3.38$$

$$\implies P(L(h) \leq \hat{L}(h) + 3.38) \geq 0.95$$

Thus the bound of 0.01 cannot be achieved on *any* dataset. But we can *construct* a dataset that could achieve this tighter bound. This could be done with kNN classifier with $k = 1$. The VC dimension of this hypothesis $= \infty$. Also, in this case $L(h) = \hat{L}(h)$ on a distribution with all the points being labeled the same class.

3 Random Forests

3.1 Normalization

The nearest neighbour classifier is affected by normalization. This is because its decision process depends on calculating distances between the different points in the data set. The classifier will assign different weights to the features if these are in different scales.

As an example of this, have a look at `ex3.py`. In this code I load the wine dataset from `sklearn` (Dheeru and Karra Taniskidou 2017) and then perform a KNN classification ($n = 3$) on the data set before normalized and then again on the normalized dataset. We notice a significant difference in the zero-one loss:

- loss without normalization 0.3333
- loss with normalization 0.0833

On the other hand, normalization will not affect a Random Forest classifier. This is because the classifier it relies on (the Tree classifiers that form a Random Forest) only cares about the *order* of the values, not about the scale. They are invariant to feature scaling. I have ran a simple Random Forest classifier (implementation from `sklearn`) on the same wine data and I get the same loss for both the default and the normalized data:

- loss of rf without normalization 0.02777
- loss of rf after normalization 0.02777

References

Abu-Mostafa, Yaser S, Malik Magdon-Ismail, and Hsuan-Tien Lin. 2012. *Learning from Data*. Vol. 4. AMLBook New York, NY, USA:

Dheeru, Dua, and Efi Karra Taniskidou. 2017. "UCI Machine Learning Repository." University of California, Irvine, School of Information; Computer Sciences. http://archive.ics.uci.edu/ml.