MLSM 2020 - Quiz 2 Solutions
08/09/2020 - 30/09/2020 at 23:59

Questions based on lecture 2

(1) (1.0 pt.)

(a) (0.5 pt.) What statement / which statements are correct about PAC learnability?

(i) The underlying distribution is fixed but unknown **True**

(ii) Generalization error bound gives the the expected generalization error on a fixed distribution generating the data **False** *The bound works on the tail of the distribution, the expected error is usually lower than what the bound indicates (lecture 2, slide 6)*

(iii) The examples should be independently drawn from an identical distribution **True**

(b) (0.5 pt.) What statement / which statements are correct about the Bayes error?

(i) For a fixed distribution generating the data, Bayes error cannot be reduced **True**

(ii) Bayes error gives the expected noise level **True**

(iii) It is possible to construct an optimal learner with a lower error than the Bayes error **False** *Bayes error is by definition the minimum achievable error (lecture 2 slide 27)*

(2) (1.0 pt.)

(a) (0.5 pt.) Based on the generalization bound relying on the size of the hypothesis class using boolean conjunctions, and the following information, what is the lower bound on the number of examples? (Formula: $m \geq \frac{1}{\epsilon}\left(\log(|\mathcal{H}|) + \log(\frac{1}{\delta})\right)$ in which the logarithms are natural.)
Dataset : 3 binary features and one binary label
Error bound : 8%
Confidence level : 96% ($\delta = 4\%$)
Note: to have the bound satisfied the fractional values should be rounded up.

(i) 82
(ii) 157
(iii) 63

Solution

$$\tfrac{1}{\epsilon}\left(\log(|\mathcal{H}|) + \log(\tfrac{1}{\delta})\right) = \tfrac{1}{0.08}\left(\log(3^3) + \log(\tfrac{1}{0.04})\right) = 81.4$$

(b) (0.5 pt.) Based on the generalization bound for true error, using the empirical error

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}},$$

if we change δ from 0.04 to 0.08, how many examples will be needed to keep the same bound as before?

(i) 0.5m
(ii) 1.6m
(iii) 0.8m

Solution To keep the same bound, the value of the square root should not change. The number of required examples, m_x, can now when δ is doubled be computed as:

$$\frac{\log(\frac{2}{\delta})}{2m} = \frac{\log(\frac{2}{2\delta})}{2m_x}.$$

Solving this and substituting $\delta = 0.04$, we get $m_x = 0.8m$ since

$$m_x = \frac{\log(\frac{2}{2\delta})}{\log(\frac{2}{\delta})} m = 0.823m$$

Questions based on lecture 3

(3) (1.0 pt.)

(a) (0.5 pt.) What statement / which statements are correct about the VC dimension?

(i) VC dimension is dependent of the training dataset **False** *VC dimension asks if any data configuration of size m can be shattered, not just subset of training set*

(ii) If VC dimension of a class of functions is m, then all possible datasets of size m can be shattered **False** *VC dimension requires that there exists at least one set that can be shattered, but not all of them need to be shattered*

(iii) VC dimension measures the ability of the classifiers from the hypothesis set to fit all the possible label configurations of at least one set of data samples of size m **True**

(b) (0.5 pt.) What statement / which statements are correct about the Rademacher complexity?

(i) Rademacher complexity can be checked empirically for a given dataset **True**

(ii) Rademacher complexity depends on the distribution generating the data **True**

(iii) Rademacher complexity measures the performance of the learning algorithm in the worst-case scenario of assigning labels to samples in adversarial way **False** *The labels are assigned randomly, not in adversarial way*

(4) (2.0 pt.) [*Computational exercise*] Consider the attached example for building a simple classification problem on the toy "two blobs" dataset. Analyse the generalization ability of a classifier whose decision function is a line (use the perceptron from sklearn; use with default parameters as shown in the example code) applied to this dataset, by plotting the training and test set errors (error can be calculated as ratio of misclassified samples to all samples), and the Rademacher and VC bounds with $\delta = 0.08$. VC-dimension of perceptron is $d + 1$, where d is the number of features.

Note: use the example code as the basis, as the random number generator is seeded there and results are the same whenever the code is run. Without this randomness is included into the results and you might not get exactly same numbers as here. The randomness in the Rademacher bound can be reduced by increasing the number of the label configurations; the variation should be small and you can choose the closest value.

(a) (1.0 pt.) How do the curves behave between $n_{tot} = 20$ and $n_{tot} = 200$ (here n_{tot} is the total number of data samples to be divided to training and testing; variable `n_tot` in the example code)? Select the correct statement/statements:

(i) Test error is always larger than training error

(ii) Rademacher generalization bound and VCdim generalization bound cross each other

(iii) The Rademacher and VC dimension bounds have similar shape, but are a little apart **True**

(b) (1.0 pt.) What are the values of Rademacher generalization bound and VCdim-based generalization bound with $n_{tot} = 100$?

(i) 1.06 and 0.96

(ii) 0.28 and 1.02

(iii) 0.64 and 1.13 **True**

(iv) 0.52 and 1.02