MLSM 2021 - Quiz 2

16/09/2022 - 28/09/2022 at 23:59

Questions based on lecture 2: Statistical learning theory

(1) (1.0 pt.) With a PAC learnable class $\mathcal{C}$, with algorithm $\mathcal{A}$ and sample S, if the generalisation error satisfies $Pr\left(R(h_S) \leq 0.1\right) \geq 0.95$, which of the statements is **true**?

(a) It is possible to run the algorithm and obtain a hypothesis with only 50% accuracy.

(b) It is not possible to obtain a hypothesis with a worse accuracy than 90%

(c) Always at least 10% of the data samples are wrongly classified.

(2) (1.0 pt.) Consider classification problem on a dataset containing 4 binary features, and a binary label. Using a rule-based classifier with boolean conjunctions, the sample complexity bound is

$$m \geq \frac{1}{\epsilon}\left(\log\left(|\mathcal{H}|\right) + \log\left(\frac{1}{\delta}\right)\right).$$

You are interested in obtaining 90% accuracy in your model. Initially you are contemplating 85% confidence as you have just enough data samples for that, but would prefer 99%. How many more samples do you need to collect to obtain the better confidence?

(a) 11 more samples

(b) 27 more samples

(c) 64 more samples

Questions based on lecture 3: Learning with infinite hypothesis classes

(3) (1.0 pt.) A classifier has a VC dimension of 4. Which of the following claims is true?

(a) Given any four data points, it is possible to shatter them with the classifier.

(b) It is not possible to shatter any collection of five data points with the classifier.

(c) It is not possible to shatter any collection of three data points with the classifier.

(4) (1.0 pt.) [*Programming exercise*] The attached example code contains a simple dataset and a binary classifier.

What is the value of the generalisation bound based on Rademacher complexity for training set sizes n=20, n=50 and n=100 (use the first n samples of the training set)? Choose the closest values (randomness from calculating the empirical Rademacher bound can result in small variation in the results). Use $\delta = 0.05$.

(a) $20 \approx 0.99$; $50 \approx 0.61$; $100 \approx 0.42$

(b) $20 \approx 1.29$; $50 \approx 0.81$; $100 \approx 0.65$

(c) $20 \approx 1.63$; $50 \approx 1.25$; $100 \approx 1.12$

(5) (1.0 pt) [*Programming exercise*] Using the same setting as the previous exercise, calculate the value of the generalisation bound based on VC-dimension. The VC-dimension of perceptron is $d + 1$, where d is the number of features.

(a) $20 = 1.21$; $50 = 0.85$; $100 = 0.64$

(b) $20 = 1.39$; $50 = 0.95$; $100 = 0.8$

(c) $20 = 1.51$; $50 = 1.05$; $100 = 0.87$

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons


# ======================================================================
# dataset

n_tot = 400
n = int(n_tot/2)
# two moons, not really linearly separable
X, y = make_moons(n_tot, noise=0.15, random_state=0)

plt.figure()
colors = ["g", "b"]
for ii in range(2):
    class_indices = np.where(y==ii)[0]
    plt.scatter(X[class_indices, 0], X[class_indices, 1], c=colors[ii])
plt.title("full dataset")
plt.show()

# divide data into training and testing
np.random.seed(42)
order = np.random.permutation(n_tot)
train = order[:n]
test = order[n:]

Xtr = X[train, :]
ytr = y[train]
Xtst = X[test, :]
ytst = y[test]

# ======================================================================
# classifier

# The perceptron algorithm will be encountered later in the course
# How exactly it works is not relevant yet, it's enough to just know it's a binary classifier
from sklearn.linear_model import Perceptron as binary_classifier

# It can be used like this:
bc = binary_classifier()
bc.fit(Xtr, ytr)  # this is how to train the classifier on training data
preds = bc.predict(Xtst)  # this is how to obtain predictions on test data
```