EE512: Machine Learning

Homework 1

Due 5:29 pm, 6th April , 2021

1. Make sure to read the questions carefully before attempting them.
2. You are allowed to discuss with fellow students and with TAs for general advice, but you must submit your own work.
3. Plagiarism will be not tolerated.
4. The write-ups must be very clear, to-the-point, and presentable. We will not just be looking for correct answers rather highest grades will be awarded to write-ups that demonstrate a clear understanding of the material. Write your solutions as if you were explaining your answer to a colleague. Style matters and will be a factor in the grade.
5. Codes and all results must be submitted on classroom as one (pdf/jupyter) file. You can use latex or take pictures/screenshots of your write-ups/code. Ideally, we would like you to submit a very well documented Jupyter notebook.

Suggested readings:

1. *Elements of Statistical Learning* (by Hastie, Tibshirani, and Friedman): Chapter 1 (pages 1-8).
2. *Learning from Data* (by Abu-Mostafa, Magdon-Ismail, Lin): Chapter 1. This covers much of the same ground we did in our "First look at generalization", and does so in a very conversational manner.
3. "Introduction to Statistical Learning Thoery" by Bousquet, Boucheron, and Lugosi: Sections 1-3 (first 13 pages). Ignore Section 2.1 for now. The beginning of this paper provides an overview of using concentration bounds to bound the performance of empirical risk minimization in the context of a finite set of hypotheses. You can download the paper from the course web page.

Problems:

1. Suppose that we have some number m of coins. Each coin has the same probability of landing on heads, denoted p. Suppose that we pick up each of the m coins in turn and for each coin do n independent coin tosses. Note that the probability of obtaining exactly k heads out of n tosses for any given coin is given by the binomial distribution:

$$\mathbb{P}[k|n,p] = \binom{n}{k} p^k (1-p)^{n-k}$$

For each series of coin tosses, we will record the results via the empirical estimates given by

$$\widehat{p}_i = \frac{\text{number of times coin } i \text{ lands on heads}}{n}$$

   (a) Assume that $n = 10$. If all the coins have $p = 0.05$, compute a formula for the exact probability that *at least* one coin will have $\widehat{p}_i = 0$. (This may be easier to calculate by instead computing the probability that this does not occur.) Give a table containing the values of this probability for the cases of $m = 1$, $m = 1,000$, and $m = 1,000,000$. Repeat for $p = 0.75$.

   (b) Now assume that $n = 10$, $m = 2$, and that $p = 0.5$ for both coins. Compute (exactly, via a formula) and then plot/sketch

$$\mathbb{P}\left[\max_i |\widehat{p}_i - p| > \epsilon\right]$$

as a function of $\epsilon \in [0,1]$. (Note that if $n = 10$, $\widehat{p}_i$ can take only 11 discrete values, so your plot will have discrete jumps at certain values as ϵ changes.) On the same plot, show the bound that results from applying the Hoeffdings inequality together with the union bound.

2. (a) Suppose that X is a Gaussian random variable with zero mean and variance σ^2:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2}$$

Find a tail bound for X using the Chernoff bounding method (see page 9 of the notes on concentration inequalities). In other words, fill in the right hand side below with an expression that depends on t (and σ^2)

$$\mathbb{P}[X > t] \leq ???$$

To make this bound as good as possible, optimize over your choice of λ. Expressions for the moment generating function of a Gaussian random variable are easy to come by (e.g., in the "Normal distribution" entry in Wikipedia).

   (b) Suppose that X_1, X_2, ..., X_m are iid Gaussian random variables with mean 0 and variance σ^2. Using your answer for part (a) and the union bound, find a bound for

$$\mathbb{P}\left[\max_{i=1,\ldots,m} X_i > t\right] \leq ???$$

(c) For X_i as in part (b), complete the following sentence: With probability at least 0.9,

$$\max_{i=1,\ldots,m} X_i \leq ???$$

(d) Using Python, create histograms for the random variable

$$Z = \max_{i=1,\ldots,m} X_i, \quad X_i \sim \text{Normal}(0,1)$$

for $m = 10^\beta$ for $\beta = 3, 4, 5, 6$. The code in `hist-example.py` should help you get started. Discuss in the context of your answer to part (c). Turn in plots of your histograms along with your comments.

3. In this problem we will explore nearest neighbor classification in Python.

The file `knn-example.py` provides a good start at this. You should be able to run this in the iPython environment simply by typing `run knn-example.py`. This uses the NumPy, Matplotlib, and scikit learn python packages. These should come included in the standard Anaconda distribution, but if you dont have them you will need to install them first.

The file begins by loading the appropriate packages and fixing the random seed so that your results will be repeatable. It then generates a simple 2-dimensional dataset with n datapoints from two possible classes. Next it builds a k-nearest neighbor classifier. Finally, it plots the results. Before going further, spend some time with this and try to understand what the code is doing.

In this problem I would like you to design a k-nearest neighbor classifier for several different values of n. In particular, I would like you to consider $n = 100, 500, 1000, 5000$. For each of these values of n, experiment with different choices of k and decide what the "best" choice of k is for each of these values of n (either based on the visual results, or using some quantitative method of your own devising). Provide a table showing your choices of k, and include a plot of the resulting classifier for each value of n.

4. Consider a binary classification problem involving a single (scalar) feature x and suppose that $X|Y = 0$ and $X|Y = 1$ are continuous random variables with densities given by

$$f_{X|Y}(x|Y=0) = g_0(x) = \frac{1}{\sqrt{2}} e^{-\sqrt{2}|x|} \quad \text{and} \quad f_{X|Y}(x|Y=1) = g_1(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

respectively.

(a) Plot g_0 and g_1.

(b) Suppose that $\pi_0 = \mathrm{P}[Y=0] = \frac{1}{2}$ and hence $\pi_1 = \mathrm{P}[Y=1] = \frac{1}{2}$. Derive the optimal classification rule in terms of minimizing the probability of error. Relate this rule to the plot of g_0 and g_1.

(c) Calculate the Bayes risk for this classification problem (i.e., calculate the probability of error for the classification rule derived above). You can use the Python to compute integrals of the Gaussian density.

5. Suppose that our probability model for (X, Y), where X takes values in $\mathbb{R}^d$ and Y takes values in $\{0, 1\}$, is given by

$$\mathrm{P}[Y=0] = \pi_0 \quad \mathrm{P}[Y=1] = \pi_1 = 1 - \pi_0$$

and the conditional densities

$$f_{X|Y}(\boldsymbol{x}|Y=0) = g_0(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_0})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu_0})}$$

$$f_{X|Y}(\boldsymbol{x}|Y=1) = g_1(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)}$$

That is, $X|Y=0$ and $X|Y=1$ are multivariate normal random variables with the same covariance matrix $\boldsymbol{\Sigma}$ and means $\boldsymbol{\mu_0}$,$\boldsymbol{\mu_1}$, respectively. (Recall that covariance matrices are symmetric and have positive eigenvalues.)

(a) Find the Bayes classification rule (in terms of the π_i, $\boldsymbol{\mu_i}$, and $\boldsymbol{\Sigma}$).

(b) Find $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that your rule can be expressed as

$$h^*(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

(This question is easier than it looks. It is really just a matter of manipulating the expressions above. Note that you can work with the log of the functions, since if $f(x) > 0$ and $g(x) > 0$,

$$f(\boldsymbol{x}) \geq g(\boldsymbol{x}) \quad \Leftrightarrow \quad \log f(\boldsymbol{x}) \geq \log(\boldsymbol{x})$$

)