Probabilistic Learning Homework

Emilio Kuhlmann, Sofya Laskina, Suresh Kumar Choudhary

February 14, 2021

Linear classification

a) Compute the Bayes decision and Bayes error for this problem

Bayes Decision:

$$\phi^*(x^1, x^2) = \begin{cases} 1 & \text{if } 1/4 \cdot (x^1+1) \cdot (x^2+1) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

Bayes error: Unfortunately we could not get the $L^* = \frac{3}{8} - \frac{\ln 2}{4}$ suggested by the hint, however by mimicking the tutorial exercise we got:

$$\mathbb{P}[Y=1] = \frac{1}{4}\int_0^2 \int_0^{\min\{2,\frac{2}{t}\}} 1dudt = \frac{1}{4}\int_0^2 \min\{2, \frac{2}{t}\}dt$$
$$= \frac{1}{4}\int_0^1 2dt + \frac{1}{4}\int_1^2 \frac{2}{t}dt = \frac{1}{2} + \frac{1}{2} \cdot \ln 2$$

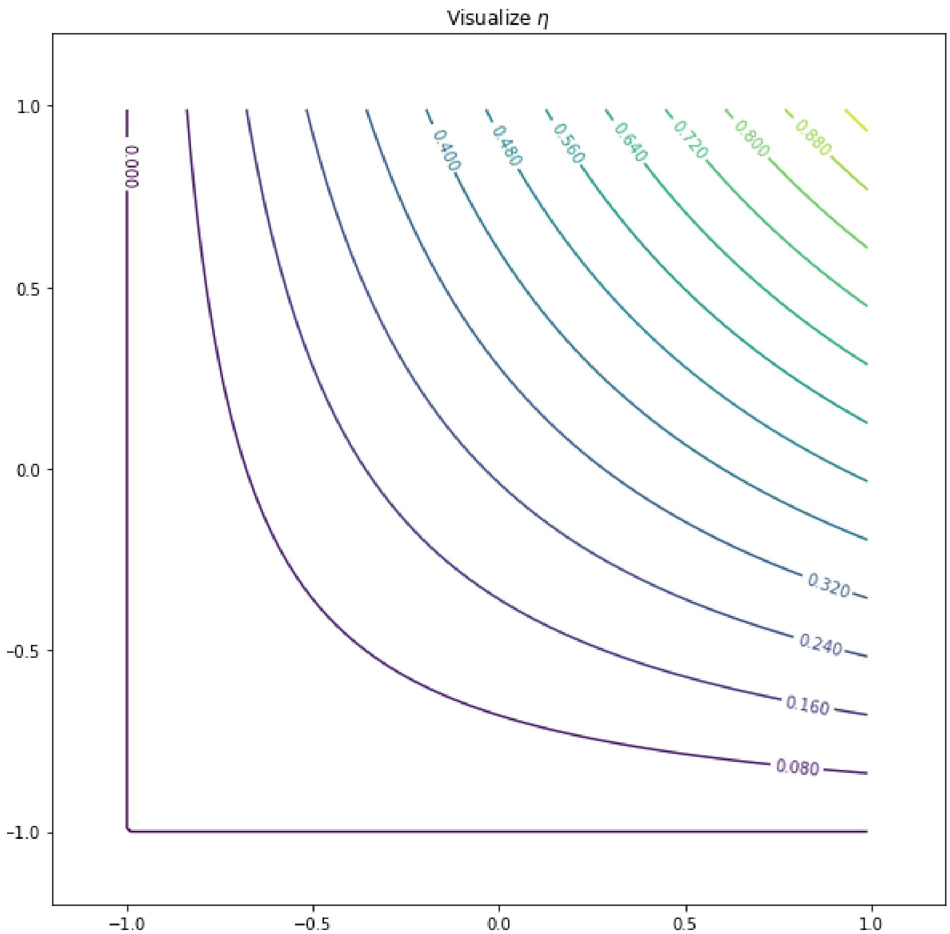
Note that we shifted this problem so that we would be sampling uniformly between 0 and 2 instead of -1 and 1. We now need to check if the product of the two values is larger than 2 instead of larger than $\frac{1}{2}$. This is equivalent to the problem stated in the homework.

Therefore:

$$L^* = \mathbb{P}[\phi^* \neq Y] = \mathbb{P}[1 \neq Y] = 1 - (\frac{1}{2} + \frac{1}{2}\ln 2) = \frac{1}{2} - \frac{1}{2}\ln 2$$

Unfortunately this is the best that we could do, any other approach led to results that weren't even close to what we were supposed to get.

b) Produce a visual estimate of the best classifier in the class of all linear decision functions



We got produced the figure by creating a meshgrid with numpy's np.meshgrid function. We then evaluated the function $\eta(x^1, x^2) = 1/4 \cdot (x^1 + 1) \cdot (x^2 + 1)$ on the grid and plotted the results with matplotlib's plt.contour function. Further details can be seen in our Jupyter notebook[1] for this week's homework.

c) Generate training data D_ℓ from the joint distribution of (X, Y) and sample the risk of the classifier.

In this section the first step was to draw ℓ samples twice (one for X^1 and one for X^2) from numpy's np.random.uniform function. Then we needed to add 1 to each entry of the draws, elementwise multiply the resulting vectors together

[1] `https://colab.research.google.com/drive/1bp8FG4biVlJSYOe8rVr4RMQWO5qYYJaE?usp=sharing`

and finally multiply them with the scalar $\frac{1}{4}$. We have now computed η. As we are interested in the Bayes' predictions ϕ^* we compute it by simply checking whether η is larger than $\frac{1}{2}$.

The next step is to draw from the distribution of Y. To do this, we draw from a Bernoulli(η_i) distribution for each of the ℓ values that η takes on.

Finally we compare the real values of Y with our predictions ϕ^* The proportion of right predictions is an approximation of L^*. With $\ell = 1000000$ we got $\widehat{L^*} = 0.202387$, very close to the analytical solution $L^* = \frac{3}{8} - \frac{\ln 2}{4}$.

d) Generate training data D_ℓ from the joint distribution of (X, Y) again. Write an algorithm that finds the empirically optimal decision function $\widehat{\phi_\ell}$ by brute force minimization of $\widehat{L}_\ell(\phi)$ over all $2 \cdot \binom{\ell}{2}$ different classifiers from the class $\mathcal{C}_\ell$ of all classifiers that are defined by lines through pairs of data points

This is similar to section c) but we do not know η.

Our approach was to use numpy's np.polyfit function to generate the coefficients for a separating hyperplane, for each pair of points from X^1 and X^2 respectively. The pair of points with the lowest empirical risk is then used to define the hyperplane that constitutes the decision boundary. Note that this is a very expensive calculation with $\mathcal{O}(\ell^3)$ operations.

Further details can be seen in our Jupyter notebook for this week's homework.

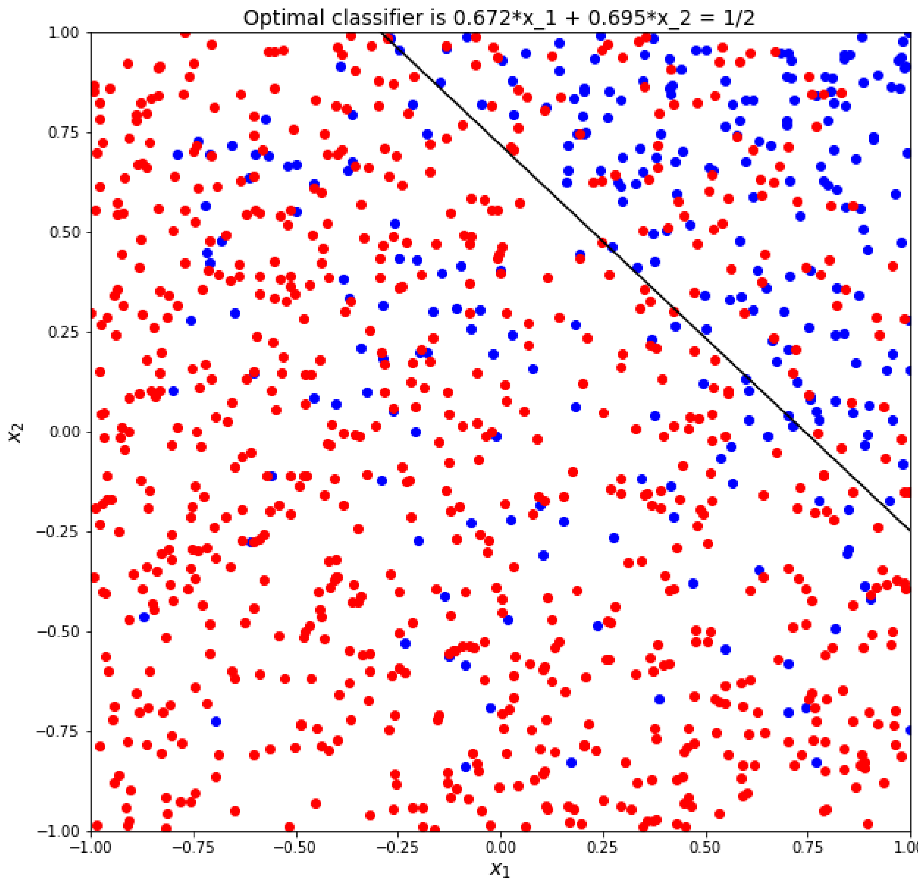
e) Estimate the generalization error of the computed classifier (i.e., the true risk $\widehat{L}_\ell(\phi)$) by sampling. Compare with (a)

With $\ell = 1000$ we got an empirical risk of 0.206. This is very close to the true risk $L^* = \frac{3}{8} - \frac{\ln 2}{4} \approx 0.2$

Consider the image below. Here we have illustrated the distribution of X^1 and X^2, and Y. The black line illustrates the separating hyperplane. The separating hyperplane satisfies the equation

$$0.672 \cdot x^1 + 0.695 \cdot x^2 = \frac{1}{2}.$$

This hyperplane describes optimal Bayes' classifier. That is good as approximating the Bayes' classifier was the goal.



f) How large should ℓ be such that the confidence level of the empirically optimal classifier being at most 10% worse than the class-optimal classifier is 0.75

According to our calculations ℓ should be at least 74907. This is the result of plugging all the values that we have into

$$\min \ell = 2/(\varepsilon)^2 \cdot \ln(2n(n-1)/\delta)$$

Where $\varepsilon = 0.1 \cdot \widehat{L}_\ell(\phi)$ and $\delta = 1 - 0.75 = 0.25$