



COLLEGE OF ENGINEERING
DANIEL GUGGENHEIM SCHOOL OF AEROSPACE ENGINEERING

ISYE 7750: MATHEMATICAL FOUNDATIONS OF MACHINE LEARNING

Homework 7

Professor:
Ashwin Pananjady
ISYE/ECE Assistant Professor

Student:
Tomoki Koike
AE PhD Student

December 1, 2022

Table of Contents

I	Problem One	2
I.i	Solution	2
I.i.1	Question (1)	2
I.i.2	Question (2)	3
II	Problem Two	4
II.i	Solution	4
III	Problem Three	6
III.i	Solution	6
III.i.1	Question (1)	6
III.i.2	Question (2)	7
IV	Problem Four	8
IV.i	Solution	8
IV.i.1	Question (1)	8
V	Problem Five	9
V.i	Solution	10
V.i.1	Question (1)	10
VI	Problem Six	11
VI.i	Solution	11
VI.i.1	Question (1)	11

I Problem One

Three friends, Aaron, Blake, and Colin, meet together every week to play poker. They each buy in for \$100, and play until one of them has it all. Poker is a game of skill, but also a game of luck — the winner each week is modeled as a discrete random variable X with distribution parameterized by θ_a and θ_b with

$$\mathbb{P}[X = A] = \theta_a, \quad \mathbb{P}[X = B] = \theta_b, \quad \mathbb{P}[X = C] = 1 - \theta_a - \theta_b,$$

where

$$\theta_a, \theta_b \geq 0, \quad \text{and} \quad \theta_a + \theta_b \leq 1. \quad (\text{I.1})$$

Above, event A corresponds to Aaron winning, B corresponds to Blake winning, and C corresponds to Colin winning.

The parameters θ_a and θ_b are unknown, and we want to infer them after observing the winners each week for many weeks. We have no idea of the relative skill of the players at the beginning of this experiment, so our prior is uniform on the triangular region specified by the constraints in (I.1):

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix}, \quad f_{\Theta}(\boldsymbol{\theta}) = \begin{cases} 2, & \boldsymbol{\theta} \in \mathcal{S}, \\ 0, & \boldsymbol{\theta} \notin \mathcal{S}, \end{cases} \quad \mathcal{S} = \{\boldsymbol{\vartheta} \in \mathbb{R}^2 : \vartheta[1], \vartheta[2] \geq 0, \vartheta[1] + \vartheta[2] \leq 1\}.$$

(You might, at this point, want to sketch the set \mathcal{S} in \mathbb{R}^2 .)

1. Show that after N weeks, where we have observed N_a wins for Aaron, N_b wins for Blake, and $N_c = N - N_a - N_b$ wins for Colin, the posterior for Θ is given by the *Dirichlet distribution*

$$f_{\Theta}(\boldsymbol{\theta} | X_1 = x_1, \dots, X_N = x_N) \propto \theta_a^{N_a} \theta_b^{N_b} (1 - \theta_a - \theta_b)^{N - N_a - N_b}.$$

(The constant in front of the expression on the right turns out to be

$$\frac{\Gamma(N+3)}{\Gamma(N_a+1)\Gamma(N_b+1)\Gamma(N-N_a-N_b+1)},$$

which is the inverse of the integral of the expression on the right over the constraint set \mathcal{S} .)

2. Using MATLAB (or Python), plot the posterior density if after a year of play, we are at

$$N_a = 5, \quad N_b = 32, \quad N_c = 15.$$

Solution

Question (1)

Computing the posterior, we have

$$\begin{aligned} f_{\Theta}(\boldsymbol{\theta} | x_1, \dots, x_N) &= \frac{f_{\mathbf{X}}(x_1, \dots, x_N | \boldsymbol{\theta}) f_{\Theta}(\boldsymbol{\theta})}{\int_{\mathcal{S}} f_{\mathbf{X}}(x_1, \dots, x_N | \boldsymbol{\theta}) f_{\Theta}(\boldsymbol{\theta}) d\boldsymbol{\theta}} \\ &= \frac{f_{\mathbf{X}}(x_1, \dots, x_N | \boldsymbol{\theta})}{\int_{\mathcal{S}} f_{\mathbf{X}}(x_1, \dots, x_N | \boldsymbol{\theta}) d\boldsymbol{\theta}} \end{aligned} \quad (\text{I.2})$$

For a single game we can say

$$f_{\mathbf{X}}(x_i | \boldsymbol{\theta}) = \{\theta_a, \theta_b, 1 - \theta_a - \theta_b\} \quad \text{for } x_i = \{A, B, C\}$$

then with conditional independence we can say for N games we have

$$f_{\mathbf{X}}(x_1, \dots, x_N | \boldsymbol{\theta}) = \theta_a^{N_a} \theta_b^{N_b} (1 - \theta_a - \theta_b)^{N - N_a - N_b}. \quad (\text{I.3})$$

Hence, we have proven that the posterior is proportional to the RHS of the expression above. ■

Question (2)

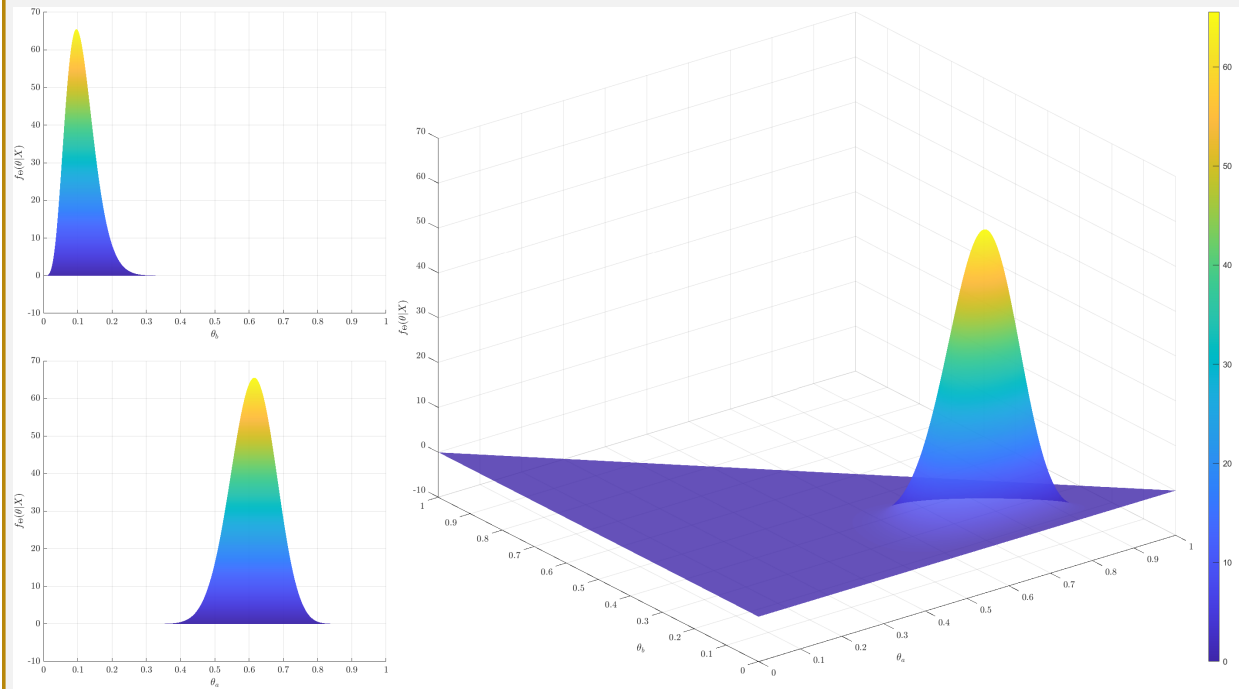


Figure 1: Posterior density plot.

II Problem Two

Suppose the random variables (X, Y) , $X \in \mathbb{R}^2$, $Y \in \{1, 2\}$, have joint distribution given by

$$\mathbb{P}[Y = 1] = \mathbb{P}[Y = 2] = 1/2,$$

$$f_X(\mathbf{x}|Y = k) = \frac{1}{2\pi\sqrt{\det(\boldsymbol{\Sigma}_k)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right),$$

where

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 3 & -6 \\ -6 & 24 \end{bmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 16 & -6 \\ -6 & 8 \end{bmatrix}.$$

Draw the regions $\Gamma_1(h^*)$ and $\Gamma_2(h^*)$ that correspond to the Bayes classifier. (You can feel free to use MATLAB or Python for this.)

Solution

Solving the probability

$$\mathbb{P}[Y = k | X = x] = \frac{\mathbb{P}[Y = k] f_X(\mathbf{x} | Y = k)}{\sum_{l=1}^K \mathbb{P}[Y = l] f_X(\mathbf{x} | Y = l)}. \quad (\text{II.1})$$

Then we know that the optimal classifier can be model as

$$\begin{aligned} h^*(\mathbf{x}) &= \operatorname{argmax}_k \mathbb{P}[Y = k | X = x] \\ &= \operatorname{argmax}_k \frac{\mathbb{P}[Y = k] f_X(\mathbf{x} | Y = k)}{\sum_{l=1}^K \mathbb{P}[Y = l] f_X(\mathbf{x} | Y = l)} \\ &= \operatorname{argmax}_k \mathbb{P}[Y = k] f_X(\mathbf{x} | Y = k) \\ &= \operatorname{argmax}_k \prod_{i=1}^D \mathbb{P}[Y = k] f_X(x_i | Y = k) \\ &= \operatorname{argmax}_k \left\{ \sum_{i=1}^D \ln[f_X(x_i | Y = k)] + \ln(\mathbb{P}[Y = k]) \right\}. \end{aligned}$$

Now we know that

$$\Gamma_1(h^*) = \{\mathbf{x} : (\mathbf{x} - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) - (\mathbf{x} - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_2^{-1}(\mathbf{x} - \boldsymbol{\mu}_2) \leq -2\kappa\}$$

where

$$\kappa = \ln \left(\frac{1/\sqrt{\det(\boldsymbol{\Sigma}_2)}}{1/\sqrt{\det(\boldsymbol{\Sigma}_1)}} \right)$$

Similarly we have

$$\Gamma_2(h^*) = \{\mathbf{x} : (\mathbf{x} - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) - (\mathbf{x} - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_2^{-1}(\mathbf{x} - \boldsymbol{\mu}_2) > -2\kappa\}$$

which is the compliment of Γ_1 . The regions can be plotted using the Monte Carlo method.

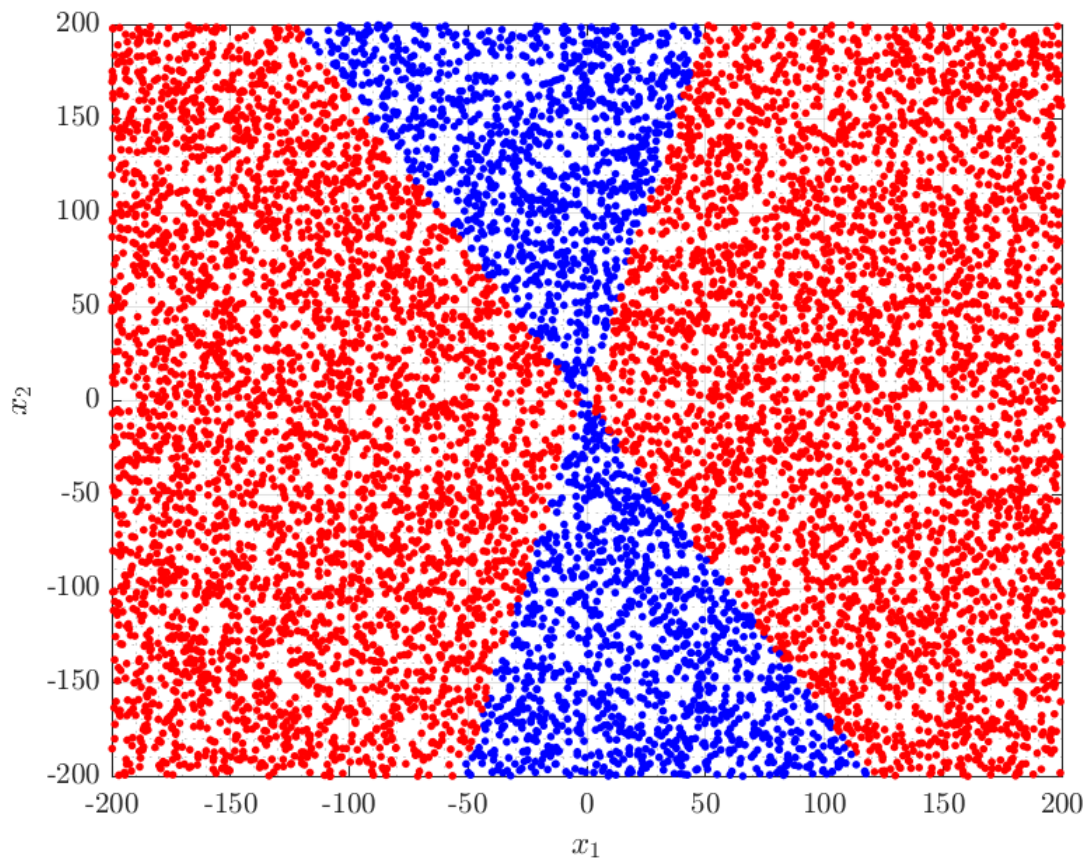


Figure 2: Region of the Bayes classification.

The blue is the region of Γ_1 and the red indicates the region of Γ_2 .

III Problem Three

1. The file `hw07p3data` contains two arrays: `X1` and `X2`. These are samples from an unknown distribution, where `X1` has been assigned “class 1”, and `X2` has been assigned “class 2”. Implement the nearest neighbor algorithm, and sketch the decision regions Γ_1 and Γ_2 that it defines.
2. Actually, the data in part (a) was generated using the model from Problem 2. Estimate the generalization error $R(h)$ for both the Bayes classifier (Problem 2) and the nearest-neighbor rule (part (a)), and compare the two. This will require the generation of many Gaussian random vectors with specified covariance matrices.

Solution

Question (1)

The region looks as follows.

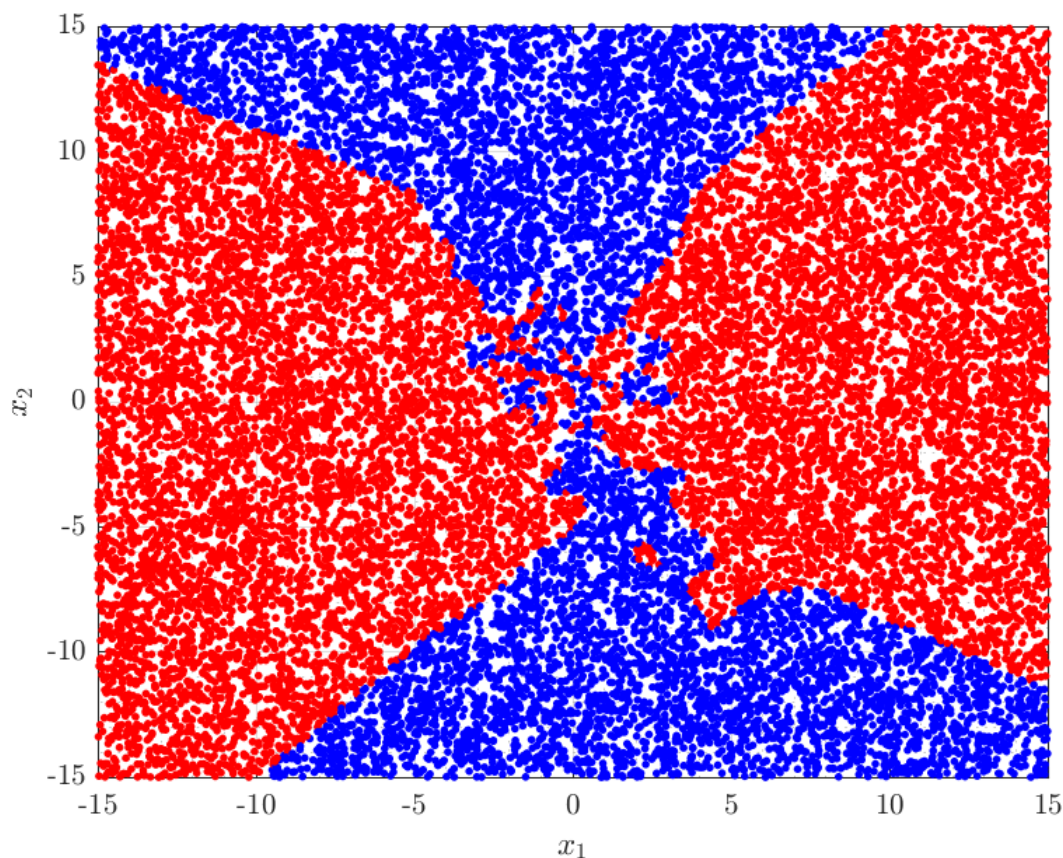


Figure 3: Region of the Nearest Neighbor classification.

Question (2)

Now if we run an empirical risk analysis by generating random points with the given mean and covariance matrix, we find that the risk of Bayes classification and the nearest neighbor algorithm are

$$R(h^*) = 0.2238, \quad R(h_{NN}) = 0.5019 \quad (\text{III.1})$$

We know that when $N \rightarrow \infty$ then $R(h_{NN}) \leq 2R(h^*)$, and our values do approximately agree with this fact.

IV Problem Four

Let X_1, X_2, \dots be independent Gaussian random variables with mean 0 and variance 1. Let

$$Z_M = \max_{1 \leq m \leq M} |X_m|.$$

- Using Monte Carlo simulation, estimate $\mathbb{E}[Z_M]$ for

$$M = 1, 2, 5, 10, 20, 50, 100, \dots, 10^5, 2 \cdot 10^5, 5 \cdot 10^5, 10^6.$$

Turn in a plot of $\mathbb{E}[Z_M]$ versus M on appropriately scaled (log) axes.

- Prove that

$$\frac{1}{\sqrt{2\pi}} \int_u^\infty e^{-t^2/2} dt \leq \frac{1}{2} e^{-u^2/2}, \quad (\text{IV.1})$$

and so

$$\mathbb{P}[|X_m| > u] \leq \min(1, e^{-u^2/2}).$$

Using this and the Boole inequality, find a bound on $\mathbb{P}[Z_M > u]$.

- Prove that if Z is a positive-valued random variable, then

$$\mathbb{E}[Z] = \int_0^\infty \mathbb{P}[Z > u] du.$$

Use this along with your answer to part (b) to get an analytical upper bound on $\mathbb{E}[Z_M]$. Note that if $f(u)$ is a positive monotonically decreasing function, then

$$\int_0^\infty \min(1, f(u)) du = \gamma + \int_\gamma^\infty f(u) du,$$

where γ is the point where $f(\gamma) = 1$.

Solution

Question (1)

V Problem Five

Suppose that the coupled random variables $(X, Y) \in \mathbb{R} \times \{0, 1\}$ have joint distribution specified by

$$\mathbb{P}[Y = 0] = 0.4, \quad X|Y = 0 \sim \mathcal{N}(-1, 4), \quad X|Y = 1 \sim \mathcal{N}(1, 4).$$

We will consider the following set of classifiers for predicting Y from an observation of X :

$$\mathcal{H} = \{h_\theta(x), \theta \in [-10, 10]\}, \quad \text{where} \quad h_\theta(x) = \begin{cases} 0, & x < \theta, \\ 1, & x \geq \theta. \end{cases}$$

In this case, because we have been told the distribution, we can compute the true risk for every $h_\theta \in \mathcal{H}$:

$$R(h_\theta) = \mathbb{P}[Y = 1] \int_{-\infty}^{\theta} f_X(x|Y = 1)dx + \mathbb{P}[Y = 0] \int_{\theta}^{\infty} f_X(x|Y = 0)dx. \quad (\text{V.1})$$

(In MATLAB/Python, you can compute the above with the help of the `normcdf/norm.cdf` command.)

1. Write code that generates N (independent) realizations of (X, Y) then plots the empirical risk function $\hat{R}_N(h_\theta)$ overlaid on top of $R(h_\theta)$. Turn in plots of three realizations each for $N = 10, 100, 1000$. These plots should have a horizontal axis indexed by $\theta \in [-10, 10]$ (and this interval should be discretized to 1000 points).
2. Using Monte Carlo simulation, estimate $\mathbb{E}[|R(h_{0.45}) - \hat{R}_N(h_{0.45})|]$ for the particular case of $\theta = 0.45$ and $N = 10, 100, 1000$. Here, the expectation is with respect to the draw of the data. For a fixed N , a single experiment consists of drawing x_1, \dots, x_N , computing $\hat{R}_N(h_{0.45})$, and then $|R(h_{0.45}) - \hat{R}_N(h_{0.45})|$ (the quantity $R(h_{0.45})$ is deterministic). Run this experiment many times and average the results to get your estimate. Then repeat for the other values of N .
3. Using Monte Carlo simulation, estimate

$$\mathbb{E}\left[\max_{h_\theta \in \mathcal{H}} |R(h_\theta) - \hat{R}_N(h_\theta)|\right]$$

for $N = 10, 100, 1000$. As above, the expectation is with respect to the random draw of the data x_1, \dots, x_N , so your simulation framework should be similar. The main difference is that every experiment produces a random *function* $\hat{R}_N(h_\theta)$ of θ that is compared against the deterministic function $R(h_\theta)$. You can compute the max by gridding the θ axis at sufficiently many points.

4. Using Monte Carlo simulation, estimate the average performance (generalization error) $\mathbb{E}[R(\hat{h}_N)]$ of the empirical risk minimizer

$$\hat{h}_N = \operatorname{argmin}_{h_\theta \in \mathcal{H}} \hat{R}_N(h_\theta),$$

for $N = 10, 100, 1000$. (You again need simulations as above to generate the \hat{h}_N — given the minimizer, computing $R(\hat{h}_N)$ can be done with (V.1).) As before, \hat{h}_N is a random classification rule (because of the randomness of the data), and so $R(\hat{h}_N)$ is a random number, even though $R(\cdot)$ is a deterministic function. Compare your estimate of $\mathbb{E}[R(\hat{h}_N)]$ to the risk of the Bayes classifier $R(h_{\text{bayes}})$, where as usual

$$h_{\text{bayes}} = \operatorname{argmin}_{h_\theta \in \mathcal{H}} R(h_\theta).$$

Solution

Question (1)

VI Problem Six

1. Compute the gradient (with respect to \mathbf{w}) of

$$-\ell(\mathbf{w}; \mathbf{x}_n, y_n) = -y_n \ln(\sigma(\mathbf{w}^\top \boldsymbol{\Psi}(\mathbf{x}_n))) - (1 - y_n) \ln(1 - \sigma(\mathbf{w}^\top \boldsymbol{\Psi}(\mathbf{x}_n)))$$

2. The file `hw07p6data.mat` contains a 2×1000 matrix \mathbf{X} and a 1×1000 binary-valued vector \mathbf{Y} . Interpret the columns of \mathbf{X} as data points $\mathbf{x}_n \in \mathbb{R}^2$ and the corresponding entry of \mathbf{Y} as a class label $y_n \in \{0, 1\}$. Implement gradient descent¹ to fit a conditional probability function to the data. For the function space \mathcal{F} , use the space of all polynomials of degree 2, that is

$$\boldsymbol{\Psi}(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{bmatrix}.$$

Plot the resulting conditional probability function $p(\mathbf{x})$ and the corresponding classification regions. Turn in these plots along with your code.

Solution

Question (1)

¹You should be able to get gradient descent to converge with a small enough fixed step size.