Math Foundations of ML, Fall 2022

Homework #7

Due Friday December 2 at 5:00pm ET

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

1. 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

$$\mathrm{P}\left(X=A\right)=\theta_a, \quad \mathrm{P}\left(X=B\right)=\theta_b, \quad \mathrm{P}\left(X=C\right)=1-\theta_a-\theta_b,$$

where

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

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 (1):

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

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

(a) 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*

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

(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 $\mathcal{S}$.)

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

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

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

$$\begin{aligned} &\mathrm{P}\left(Y=1\right) = \mathrm{P}\left(Y=2\right) = 1/2, \\ &f_X(\boldsymbol{x}|Y=k) = \frac{1}{2\pi\sqrt{\det(\boldsymbol{\Sigma}_k)}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right), \end{aligned}$$

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.)

3. (a) 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.

   (b) 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.

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

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

   (a) Using Monte Carlo simulation, estimate $\mathrm{E}[Z_M]$ for

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

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

   (b) Prove that

$$\frac{1}{\sqrt{2\pi}} \int_u^\infty e^{-t^2/2}\, dt \;\le\; \frac{1}{2} e^{-u^2/2}, \tag{2}$$

   and so

$$\mathrm{P}\left(|X_m| > u\right) \;\le\; \min\left(1, e^{-u^2/2}\right).$$

   Using this and the Boole inequality, find a bound on $\mathrm{P}\left(Z_M > u\right)$.

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

$$\mathrm{E}[Z] = \int_0^\infty \mathrm{P}\left(Z > u\right)\, du.$$

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

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

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

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

$$\mathrm{P}\left(Y=0\right) = 0.4, \quad X|Y=0 \sim \text{Normal}(-1, 4), \quad X|Y=1 \sim \text{Normal}(1, 4).$$

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

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

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

$$R(h_\theta) = \mathrm{P}\left(Y=1\right) \int_{-\infty}^{\theta} f_X(x|Y=1)dx \;+\; \mathrm{P}\left(Y=0\right) \int_{\theta}^{\infty} f_X(x|Y=0)dx. \tag{3}$$

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

(a) 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).

(b) Using Monte Carlo simulation, estimate $\mathrm{E}[|R(h_\theta) - \hat{R}_N(h_\theta)|]$ 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 $x_1, \ldots, x_N$, 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.

(c) Using Monte Carlo simulation, estimate

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

for $N = 10, 100, 1000$. As above, the expectation is with respect to the random draw of the data $x_1, \ldots, x_N$, 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.

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

$$\hat{h}_N = \arg\min_{h_\theta \in \mathcal{H}} \hat{R}_N(h_\theta),$$

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 (3).) 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 $\mathrm{E}[R(\hat{h}_N)]$ to the risk of the Bayes classifier $R(h_{\text{bayes}})$, where as usual

$$h_{\text{bayes}} = \arg\min_{h_\theta \in \mathcal{H}} R(h_\theta).$$

6. (a) Compute the gradient (with respect to $\boldsymbol{w}$) of

$$-\ell(\boldsymbol{w}; \boldsymbol{x}_n, y_n) = -y_n \log(\sigma(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Psi}(\boldsymbol{x}_n))) - (1 - y_n) \log(1 - \sigma(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Psi}(\boldsymbol{x}_n)))$$

(b) The file `hw07p6data.mat` contains a 2×1000 matrix `X` and a 1×1000 binary-valued vector `Y`. Interpret the columns of `X` as data points $\boldsymbol{x}_n \in \mathbb{R}^2$ and the corresponding entry of `Y` as a class label $y_n \in \{0, 1\}$. Implement gradient descent[1] 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

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

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

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