Akash Patel
MFML-HMWK 10

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

$$P(Y=1) = P(Y=2) = \tfrac{1}{2} \quad f_X(x \mid Y=k) = \frac{1}{2\pi\sqrt{|\Sigma_k|}} \exp\left(-\tfrac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right),$$

where $\mu_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 3 & -6 \\ -6 & 24 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 16 & -6 \\ -6 & 8 \end{bmatrix}$

Draw the regions $\Gamma_1(h^*)$ and $\Gamma_2(h^*)$ that correspond to the Bayes classifier.

$\Gamma_1(h^*) = \{x : h^*(x) = 1\}$ $\qquad$ $\Gamma_2(h^*) = \{x : h^*(x) = 2\}$

$$h^*(x) = \underset{k \in \{1,\dots,K\}}{\arg\max}\ P(Y=k \mid X=x) = \underset{k \in \{1,\dots,K\}}{\arg\max} \frac{P(X=x \mid Y=k)P(Y=k)}{P(X=x)} = \underset{k \in \{1,\dots,K\}}{\arg\max}\ P(X=x \mid Y=k)P(Y=k)$$

3. a) The file hw10p3data 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) In actuality, the data in the last part 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 $E[Z_m]$ for $M = 1, 2, 5, 10, 20, 50, 100, \ldots, 10^5, 2\cdot10^5, 5\cdot10^5, 10^6$. Turn in a plot of $E[Z_M]$ versus M on appropriately scaled (log) axes.

b) It is a fact that

$$\frac{1}{\sqrt{2\pi}} \int_u^\infty \exp(-t^2/2)\,dt \le \frac{1}{2}\exp(-u^2/2),$$

and so $P(|X_m| > u) \le \min(1, e^{-u^2/2})$,

for the $X_m \sim \mathrm{Normal}(0,1)$ as above. Using this and the Boole inequality, find a bound on $P(Z_M > u)$.

Boole Inequality: $P\left(\bigcup_{m=1}^{M} A_m\right) \le \sum_{m=1}^{M} P(A_m)$

For our case A_m corresponds to $|X_m| > u$

$$\Rightarrow P\left(\bigcup_{m=1}^{M} |X_m| > u\right) \le \sum_{m=1}^{M} P(|X_m| > u)$$

$$\le \sum_{m=1}^{M} \min(1, e^{-u^2/2})$$

$$\boxed{P(Z_M > u) \le \min(1, M e^{-u^2/2})}$$

c) It is also a fact that if Z is a positive-valued random variable, then

$$E[Z] = \int_0^\infty P(Z > u)\,du.$$

Use this along with your answer to part (b) to get an analytical upper bound on $E[Z_M]$. 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 $f(\gamma) = 1$. You will find that fact handy along with another application of (1).

$$E[Z_m] = \int_0^\infty P(Z_m > u)\,du \leq \int_0^\infty \min(1, Me^{-u^2/2})\,du$$

$$= \sqrt{\ln(M^2)} + \int_{\sqrt{\ln(M^2)}}^\infty Me^{-u^2/2}$$

$$\leq \sqrt{\ln(M^2)} + M\frac{\sqrt{2\pi}}{2}\exp(-(\sqrt{\ln(M^2)})^2/2)$$

$$= \boxed{\sqrt{\ln(M^2)} + \frac{\sqrt{2\pi}}{2}}$$

$$E[Z_m] \leq \sqrt{\ln(M^2)} + \frac{\sqrt{2\pi}}{2}$$

$Me^{-u^2/2} = 1$

$e^{-u^2/2} = \frac{1}{M}$

$\ln(e^{-u^2/2}) = \ln(\frac{1}{M})$

$-u^2/2 = \ln(\frac{1}{M})$

$-u^2 = 2\ln(\frac{1}{M})$

$u^2 = -2\ln(\frac{1}{M})$

$u^2 = 2\ln(M)$

$u^2 = \ln(M^2)$

$u = \sqrt{\ln(M^2)}$

$Me^{-u^2/2} = 1$

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

$$\int_u^\infty e^{-t^2/2}\,dt \leq \frac{\sqrt{2\pi}}{2}e^{-u^2/2}$$

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

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

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

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

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

$$R(h_\theta) = P(Y=1)\int_{-\infty}^{\theta} f_X(x|Y=1)\,dx + P(Y=0)\int_{\theta}^{\infty} f_X(x|Y=0)\,dx$$

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 $E[|R(h_\theta)-\hat{R}_N(h_\theta)|]$ 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

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

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) $E[R(\hat{h}_N)]$ 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 (2).) 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 $E[R(\hat{h}_N)]$ to the risk of the Bayes classifier $R(h_{Bayes})$, whee as usual

$$h_{Bayes} = \arg\min_{h_\theta \in \mathcal{H}} R(h_\theta)$$

6. a) Compute the gradient (with respect to $w \in \mathbb{R}^p$) of

$$-\ell(w; x_n, y_n) = -y_n \log(\sigma(w^T \Psi(x_n))) - (1-y_n)\log(1-\sigma(w^T \Psi(x_n)))$$

$$\frac{\partial(-\ell(w; x_n y_n))}{\partial w} = \frac{\partial(-y_n \log(\sigma(w^T \Psi(x_n))) - (1-y_n)\log(1-\sigma(w^T \Psi(x_n))))}{\partial w}$$

$$= -y_n \frac{\partial(\log(\sigma(w^T \Psi(x_n))))}{\partial w} - (1-y_n)\frac{\partial(\log(1-\sigma(w^T \Psi(x_n))))}{\partial w}$$

$$= -y_n \frac{1}{\sigma(w^T\Psi(x_n))} \frac{\partial(\sigma(w^T\Psi(x_n)))}{\partial w} - (1-y_n)\frac{1}{1-\sigma(w^T\Psi(x_n))} \frac{\partial(1-\sigma(w^T\Psi(x_n)))}{\partial w}$$

$$= -y_n \frac{1}{\sigma(w^T\Psi(x_n))} \frac{\partial(\sigma(w^T\Psi(x_n)))}{\partial w} + (1-y_n)\frac{1}{1-\sigma(w^T\Psi(x_n))} \cdot \frac{\partial(\sigma(w^T\Psi(x_n)))}{\partial w}$$

$$\frac{\partial\sigma(w^T\Psi(x_n))}{\partial w} = \sigma(w^T\Psi(x_n))(1-\sigma(w^T\Psi(x_n)))\frac{\partial(w^T\Psi(x_n))}{\partial w} = \sigma(w^T\Psi(x_n))(1-\sigma(w^T\Psi(x_n)))\Psi(x_n)$$

$$= -y_n \cancel{\frac{1}{\sigma(w^T\Psi(x_n))}\sigma(w^T\Psi(x_n))}(1-\sigma(w^T\Psi(x_n)))\Psi(x_n) + (1-y_n)\cancel{\frac{1}{1-\sigma(w^T\Psi(x_n))}}\sigma(w^T\Psi(x_n))\cancel{(1-\sigma(w^T\Psi(x_n)))}\Psi(x_n)$$

$$= (-y_n(1-\sigma(w^T\Psi(x_n))) + (1-y_n)\sigma(w^T\Psi(x_n)))\Psi(x_n)$$

$$= (-y_n + \cancel{y_n\sigma(w^T\Psi(x_n))} + \sigma(w^T\Psi(x_n)) - \cancel{y_n\sigma(w^T\Psi(x_n))})\Psi(x_n)$$

$$\boxed{= (\sigma(w^T\Psi(x_n)) - y_n)\Psi(x_n)}$$

b) The file hw10ptdata.mat contains a 2x1000 matrix X and a 1x1000 binary-valued vector Y. Interpret the columns of X as data points $x_n \in \mathbb{R}^2$ and the corresponding entry of Y as a class label $y_n \in \{0,1\}$. Implement gradient descent 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

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

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