Machine Learning - Home assignment 2

Denis Trebula - (jmp640)

December 5, 2017

Section 1. Illustration of Hoeffding's Inequality

In this section are resolved problems 1.1 up to 1.7, also part of this section are two depictions of calculation from each questions. The two graphs are kept on a separate page in order to keep everything intact.

Question 1.1

In order to resolve this problem we need to make a implementation of the coin flips. The implementation is simple function which uses repeated bernoulli distribution with $\mu = 0.5, n = 20$ times 1000000. Which can resolve in a huge 20 x 1,000,000 matrix.

Question 1.2

On the next page is the plot of the frequencies of observing $P(\sum X \geq \alpha), \alpha \in [0.5, 0.55, \ldots, 0.95, 1]$ and $N = 20$. Depiction shows the probability decreases as the right side of a normal distribution. Central limit theorem ensures that the parameter space θ is normally distributed. Here it's noted that θ is the empirical average.

Question 1.3

It is not mandatory to add any more atomicity/granularity to the distribution of alpha because the empirical average of 20 coin flips cannot take any discrete value between the given alphas.

E.g. let's say we have 20 coin flips with equal heads and equal tails. We know that the empirical average will be 0.5 which is simple, so let's say that in the following 20 coin flips there will be unequal distribution of heads and tails, for example: 9 heads and 11 tails. Case like this will achieve empirical average of 0.55, which imply a granularity of 0.05

Question 1.6

Following three plots in Figure 2 show the empirical frequency, Markov's and Hoeffding's bound for the case of one million repetitions of the experiment of drawing 20 independent and indentically distributed random variable. We can observe that the Hoeffding's bound gives a tighter bound than Markov's bound. Moreover, the Hoeffding's bound plot follows relatively well with the plot of the empirical frequency, which means that it is an indication of the true probability

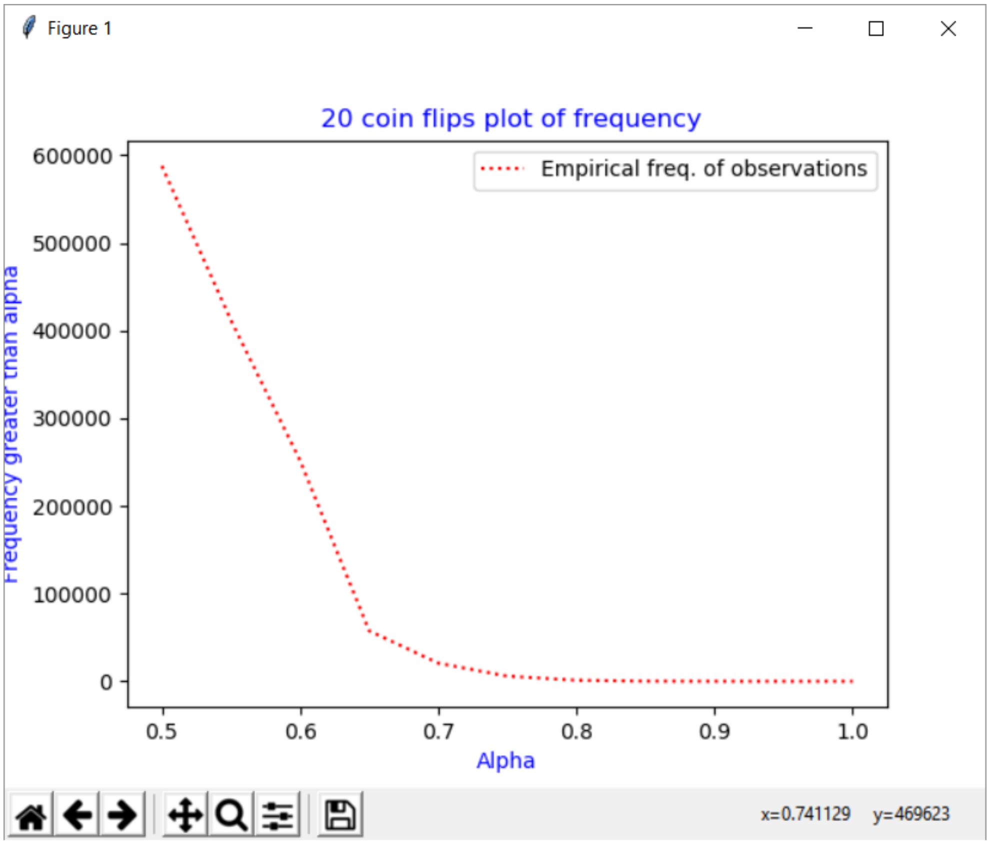

Figure 1: Coin flips

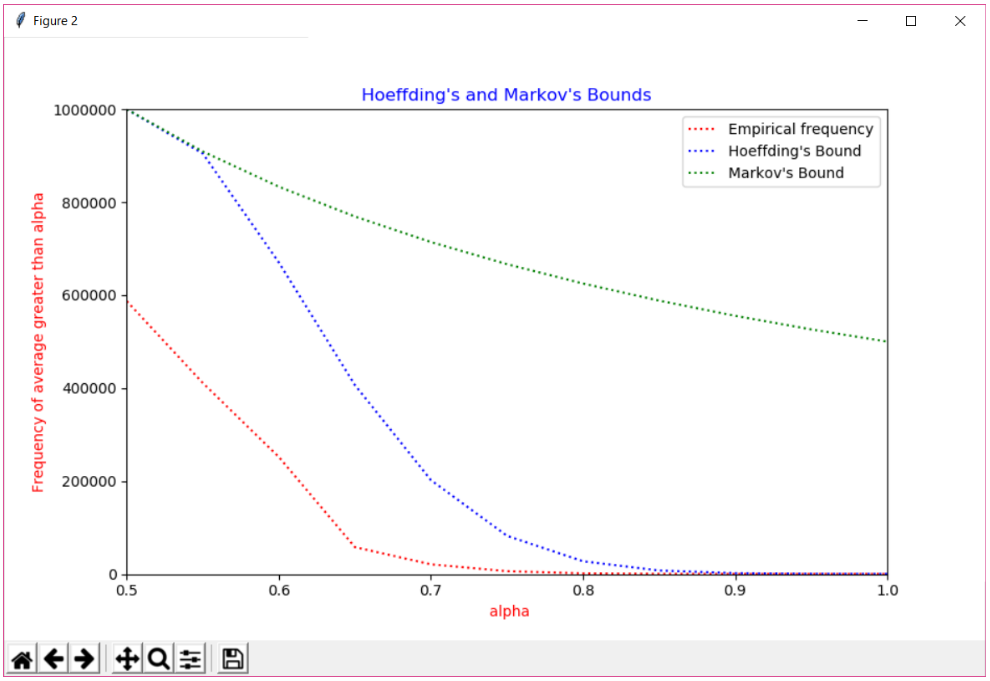

Figure 2: Distributions of Z

Question 1.7

To calculate the exact probability of $N = 20$ we need to do following equations:

$$
\begin{aligned}
P\left\{\sum X_i \geq= 1\right\} &= \prod_{i=1}^{N} \frac{1}{2} = 9.537 \times 10^{-7} \\
P\left\{\sum X_i \geq= 0.95\right\} &= \sum_{\alpha \in \{0.95, 1\}} P(X \geq \alpha) = 2.003 \times 10^{-5} \\
MarkovsBounds(\alpha = 1.00, \mu = 0.5, N = 20) &= 0.5 \\
MarkovsBounds(\alpha = 1.00, \mu = 0.5, N = 20) &= 0.526 \\
HoeffdingsBounds(\alpha = 1.00, \mu = 0.5, N = 20) &= 4.540 \times 10^{-5} \\
HoeffdingsBounds(\alpha = 0.95, \mu = 0.5, N = 20) &= 0.0003035
\end{aligned}
$$

Section 2. The effect of scale (range) and normalization of random variables in Hoeffding's inequality

We begin by substituting ε with $n\varepsilon$ in `Theorem 2.2`:

$$\mathbb{P}\left\{\sum_{i=1}^{n} X_i - \left[\sum_{i=1}^{n} X_i\right] \geq n\varepsilon\right\} \leq e^{-2(n\varepsilon)^2 / \sum_{i=1}^{n}(b_i - a_i)^2}$$

Now by looking at the right side of the inequality, in the corollary that we have to prove following X_i we can only assume values 0 and 1 because of:

$$\sum_{i=1}^{n}(b_i - a_i)^2 = n$$

Thus the right side of the inequality can be trimmed to:

$$e^{-2n\varepsilon^2}$$

Now we see left side so we can divide by n on both sides of the inequality which will give us following equation:

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{1}{n}\left[\sum_{i=1}^{n} X_i\right] \geq \frac{1}{n} n\varepsilon\right\}$$

By linearity of expectation and the assumption of corollarity of 2.4 we substitute μ instead of the expected value. Combining everything so far we achieve the result of:

$$\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq \varepsilon\right\} \leq e^{-2n\varepsilon^2}$$

Section 3. Probability in Practice

This part of assignment consists of solving two questions regarding probabilities, Hoeffding's inequality and so on.

3.1

We want to bound the probability of the problem. For this problem we will use Hoeffding's inequality. First we need to know that for each passenger there is exactly a $\mu = 1 - 0.05$ chance of him not showing up. We will also say that $n(\mu + \epsilon) = 100$ and $n = 100$. As before we solve for ϵ:

$$100(\epsilon + \frac{95}{100}) = 100, \textit{ yields: } \epsilon = \frac{1}{20}$$

The value for ϵ can then be put into the following inequality below:

$$Pr\left[\sum_{i=1}^{n} X_i \geq 100\right] \leq e^{-2\times 100 \times \frac{1}{20}^2} = e^{-\frac{1}{2}} \approx 0.60$$

Where $e^{-\frac{1}{2}} \approx 0.60$ is our result.

3.2

Establishing bound such as described in this current problem is a two step operation. First of all we need to form a bound on the reliability of the estimate of the mean. Therefore we need to bound the probability that the mean of the 10000 samples is an underestimate of the true mean. This is important as an underestimate would imply that more people will show up than we expected. This will result in a greater loss. If we set the $\varepsilon = \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}$ and substitute it into Hoeffding's inequality, we get the following equations:

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=i}^{n} X_i - \mu \geq \varepsilon\right) \leq e^{-2n\varepsilon^2} &= \mathbb{P}\left(\frac{1}{n}\sum_{i=i}^{n} X_i - \mu \geq \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) \leq e^{-2n\sqrt{\frac{\ln\frac{1}{\omega}}{2n}}^{2}} \\ &= \mathbb{P}\left(\frac{1}{n}\sum_{i=i}^{n} X_i - \mu \geq \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) \leq \omega \\ &= \mathbb{P}\left(\frac{1}{n}\sum_{i=i}^{n} X_i \geq \mu + \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) \leq \omega\end{aligned}$$

This is the calculated probability that the estimated mean is greater than the true mean and we can take in account that our choice of epsilon is bounded by ω.

Section 4. Logistic regression

This part of the assignment consists of 4 questions, each resolved in the subsections below.

4.1 Cross-entropy error measure

Here we will solve Cross entropy error measure exercise 3.6 on page 92 in the course book which consists of part (a) and part (b).

4.1.1 (a)

We have the likelihood function described as following:

$$Pr\{\, y \,|\, \mathbf{x} \,\} = \begin{cases} h(\mathbf{x}) & \text{for } y = +1 \\ 1 - h(\mathbf{x}) & \text{for } y = -1 \end{cases} \tag{1}$$

We are aware of the fact that the maximum likelihood selects the hypothesis h, which maximizes the probability, which is equivalent to minimizing the quantity:

$$\frac{1}{N}\sum_{n=1}^{N} ln\left(\frac{1}{Pr\,(\,y\,|\,\mathbf{x}\,)}\right) \tag{2}$$

We can then edit (1) in terms of indicator variables into the following equation:

$$Pr\,\{\,y\,|\,\mathbf{x}\,\} = \mathbb{1}_{y\in\{+1\}}h(\mathbf{x}) + \mathbb{1}_{y\in\{-1\}}(1-h(\mathbf{x})) \tag{3}$$

If we now edit step (2) in terms and laws of step (3) we will obtain the following equation:

$$\frac{1}{N}\sum_{n=1}^{N} ln\left(\frac{1}{Pr\,(\,y\,|\,\mathbf{x}\,)}\right) = \frac{1}{N}\sum_{n=1}^{N} ln\left(\frac{1}{\mathbb{1}_{y\in\{+1\}}h(\mathbf{x}) + \mathbb{1}_{y\in\{-1\}}(1-h(\mathbf{x}))}\right) \tag{4}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{y\in\{+1\}} ln\left(\frac{1}{h(\mathbf{x})}\right) + \mathbb{1}_{y\in\{-1\}} ln\left(\frac{1}{1-h(\mathbf{x})}\right) \tag{5}$$

We can now see that this concludes the proof.

4.1.2 (b)

Now in this question we need to prove that minimizing the in-sample error from the question **4.1.1 (a)** is equal to minimizing the following in-sample error below:

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right) \tag{6}$$

when $h(x) = \theta(\mathbf{w}^T\mathbf{x}) = \dfrac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}} = \dfrac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$

If we alternate this definition of $h(x)$ into the step (5) we will get the following equation:

$$\mathbb{1}_{y\in\{+1\}} ln\left(\frac{1}{\theta(\mathbf{w}^T\mathbf{x})}\right) + \mathbb{1}_{y\in\{-1\}} ln\left(\frac{1}{1-\theta(\mathbf{w}^T\mathbf{x})}\right) \tag{7}$$

From the second term we sign that: $\theta(-x) = \dfrac{e^{-x}}{1+e^{-x}} = \dfrac{1}{1+e^{s}} = 1-\theta(x)$
thus $1-\theta(x) = \theta(-x)$
We can now write following equation:

$$\mathbb{1}_{y\in\{+1\}} ln\left(\frac{1}{\theta(\mathbf{w}^T\mathbf{x})}\right) + \mathbb{1}_{y\in\{-1\}} ln\left(\frac{1}{\theta(-\mathbf{w}^T\mathbf{x})}\right) = ln\left(\frac{1}{\theta(y_i\mathbf{w}^T\mathbf{x})}\right) \tag{8}$$

Formulating the sigmoid function we get following equation:

$$ln\left(\frac{1}{\theta(y_i\mathbf{w}^T\mathbf{x})}\right) = ln\left(\frac{1}{\dfrac{1}{1+e^{-y_n\mathbf{w}^T\mathbf{x}_n}}}\right) \tag{9}$$

$$= ln\left(1 + e^{e^{-y_n\mathbf{w}^T\mathbf{x}_n}}\right) \tag{10}$$

therefore we get the desired in-sample error which is:

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right) \tag{11}$$

4.2 Logistic regression loss gradient

Here we will solve exercise on logistic regression loss gradient, 3.7 on page 92 in course book.

First we assume that in-sample error measure is defined as following equation:

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right) \tag{12}$$

Next step is determining the gradient of the in-sample loss error measure which corresponds to:

$$\nabla E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial}{\partial \mathbf{w}}\left[ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right)\right] \tag{13}$$

If we let $f(x) = ln(x)$ and $g(x) = 1 + e^{-y_n \mathbf{w}^T \mathbf{x_n}}$, we apply the chain rule for gradients which means that we will get:

$$\frac{\partial}{\partial \mathbf{w}}\left[ln\left(1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right)\right] = f'(g(\mathbf{w}))\nabla(\mathbf{w}) \tag{14}$$

Applying above equation we get following:

$$f'(g(\mathbf{w})) = \frac{1}{1 + e^{-y_n \mathbf{w}^T \mathbf{x}_i}} \tag{15}$$

$$\nabla g(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}}\left[1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}\right] = e^{-y_n \mathbf{w}^T \mathbf{x}_n} \times (-y_n \mathbf{x}_n) \tag{16}$$

Now we can calculate following:

$$\begin{aligned} f'(g(\mathbf{w}))\nabla(\mathbf{w}) &= \frac{-y_n \mathbf{x}_n e^{-y_n \mathbf{w}^T \mathbf{x}_n}}{1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}} &(17)\\ &= \frac{-y_n \mathbf{x}_n e^{-y_n \mathbf{w}^T \mathbf{x}_n}/e^{-y_n \mathbf{w}^T \mathbf{x}_n}}{1 + e^{-y_n \mathbf{w}^T \mathbf{x}_n}/e^{-y_n \mathbf{w}^T \mathbf{x}_n}} &(18)\\ &= \frac{-y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}} &(19)\end{aligned}$$

Based on the above equations we can say that:

$$\nabla E_{in}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} \frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}} \tag{20}$$

Now it is simple and clear that:

$$\frac{1}{N}\sum_{n=1}^{N} -y_n \mathbf{x}_n \theta(-y_n \mathbf{w}^T \mathbf{x}_n) = -\frac{1}{N}\sum_{n=1}^{N} \frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}}$$

If we write out the sigmoid function we will get the following equations below:

$$\begin{aligned} \frac{1}{N}\sum_{n=1}^{N} -y_n \mathbf{x}_n \theta(-y_n \mathbf{w}^T \mathbf{x}_n) &= \frac{1}{N}\sum_{n=1}^{N} -y_n \mathbf{x}_n \frac{1}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}}\\ &= \frac{1}{N}\sum_{n=1}^{N} \frac{-y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}}\\ &= -\frac{1}{N}\sum_{n=1}^{N} \frac{y_n \mathbf{x}_n}{1 + e^{y_n \mathbf{w}^T \mathbf{x}_n}}\end{aligned}$$

Which completes the proof needed.

4.3 Logistic Regression Implementation + 4.4 Iris Flower Data

I have successfully implemented the logistic regression classifier in `Python` language. I have used libraries such as `Pandas`, `Numpy` and `matplotlib` in many cases which allowed us to make a competitive implementation of logistic regression.
Every function is implemented in a single script Assign2.py. To reproduce results it is possible to run mentioned script using ”python3 Assign2.py” from the terminal or just simply running it in some development environment such as Pycharm. In order to successfully obtain the same results it is required to have libraries mentioned libraries. They can be easily downloaded by running command ’pip install numpy’ in terminal with their names.

The implementation of logistic regression uses the ”steepest descent” approach. Building the affine linear model gives the parameters shown below. Using these parameters to perform a classification yields the zero to one losses that are also mentioned below.

Variables from affine linear model and loss	
weights	[0.68113276 , -2.44553796]
b	-2.94487779088
Training data loss	9.6774 %
Test data loss	7.6923 %

Here are some brief examples of implementation as recommended.

```
#Function for calculating whole gradient
def gradient(dataX, dataY, weights):
  accum = initWeights(dataX)
  n = len(dataY)
  for i in range(len(dataX)):
    gradient = gradOneSimple(dataX[i], dataY[i], weights)
    accum += gradient
    mean = np.divide(accum, n)
    gradient = np.negative(mean)
  return gradient

# Function for logistic regression using LFD algorithm.
def logReg(dataX, dataY, learningRate):
  X = bigX(dataX)
  weights = initWeights(X)
  for i in range(0,1000):
    g = gradient(X, dataY, weights)
    direction = -g
    weights = updateWeights(weights, direction, learningRate)
  return weights

# Function that finds the number of wrong classifications for testing data.
def testingFalse(trainX, trainY, testX, testY, learningRate):
  trueCount = 0
  falseCount = 0
  weights = logReg(trainX, trainY, learningRate)
  vectorW = weights[:-1]
  b = weights[-1]
  for i in range(0,len(testY)):
    if linearClassifier(testX[i], vectorW, b) == testY.item(i):
      trueCount += 1
    else:
      falseCount += 1
  return falseCount
```