LUMS
A Not-for-Profit University

Department of Electrical Engineering
School of Science and Engineering

EE514/CS535 Machine Learning

HOMEWORK 2 – SOLUTIONS

Due Date: 23:55, Friday, March 26, 2021 (Submit online on LMS)
Format: 5 problems, for a total of 80 marks
Contribution to Final Percentage: 2.5%
Instructions:

- Each student must submit his/her own hand-written assignment, scanned in a single PDF document.
- You are allowed to collaborate with your peers but copying your colleague's solution is strictly prohibited. Anybody found guilty would be subjected to disciplinary action in accordance with the university rules and regulations.
- Note: Vectors are written in lowercase and bold in the homework, for your written submission kindly use an underline instead. In addition, use capital letters for matrices and lowercase for scalars.

Problem 1 (10 marks)

Bayes' Theorem - Suppose that you have built an email classifier that filters emails as 'spam' and 'not spam'. It is estimated that 80% of all emails are not spam. You claim that your classifier accurately predicts spam 90% of the time, and only incorrectly predicts as spam 1% of the time.

Using the information provided, deduce the probability of a non-spam email, given that it has been classified as spam by your classifier.

Solution: Let:

$$\begin{aligned} N &: \text{Event that an email is not spam} \\ S &: \text{Event that an email is spam} \\ C^N &: \text{Event that an email is classified as not spam} \\ C^S &: \text{Event that an email is classified as spam} \end{aligned}$$

We know:

$$\begin{aligned} P(N) &= 0.8 \\ P(S) &= 0.2 \\ P(C^S|S) &= 0.9 \\ P(C^S|N) &= 0.01 \end{aligned}$$

We need to find:

$$P(N|C^S) = \frac{P(C^S|N)P(N)}{P(C^S)}$$

Using Law of Total Probability:

$$\begin{aligned} P(C^S) &= P(C^S|S)P(S) + P(C^S|N)P(N) \\ &= 0.9 * 0.2 + 0.01 * 0.8 \\ &= 0.188 \end{aligned}$$

Plugging back into Bayes' Theorem:

$$\begin{aligned} P(N|C^S) &= \frac{P(C^S|N)P(N)}{P(C^S)} \\ &= \frac{0.01 * 0.8}{0.188} \\ &= 0.0425 \end{aligned}$$

Problem 2 (15 marks)

Maximum Likelihood Estimation (MLE) of Poisson Distribution - The probability mass function (PMF) of a random variable X that follows the Poisson distribution is given by:

$$p_X(x, \lambda) = \frac{e^{-\lambda}\lambda^x}{x!}$$

where x is a non-negative integer, i.e., $x \in \mathbb{Z}^+$ and λ, which parameterizes the distribution, is the mean of the distribution.

You are provided n independent observations of this Poisson distribution. Formulate MLE problem for the estimation of parameter λ and prove that it is simply given by the sample mean of the observations, that is:

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Hint: Follow the same steps as the example given in slides, take log likelihood function of the PMF, then derivate it with respect to λ and equate to zero.

Solution: Since we have n independent observations, we can simply take the product of probability mass functions to get the likelihood function:

$$L(\lambda, \boldsymbol{x}) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$$

where $\boldsymbol{x}$ contains the n observations.
Next we take the natural logarithm of the likelihood function on both sides to get the log likelihood function.

$$\begin{aligned}
l(\lambda, \boldsymbol{x}) &= \ln \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \\
&= \sum_{i=1}^{n} \ln \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \\
&= \sum_{i=1}^{n} \ln e^{-\lambda} + \ln \lambda^{x_i} - \ln(x_i!) \\
&= \sum_{i=1}^{n} \ln \lambda^{x_i} - \ln(x_i!) - \lambda \\
&= -n\lambda + \ln\lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \ln(x_i!)
\end{aligned}$$

To find the value of λ that maximizes this log likelihood function, we'll take the derivative with respect to λ:

$$\begin{aligned}
\frac{d}{d\lambda} l(\lambda, \boldsymbol{x}) &= \frac{d}{d\lambda}(-n\lambda + \ln\lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \ln(x_i!)) \\
&= -n + \frac{1}{\hat{\lambda}} \sum_{i=1}^{n} x_i = 0 \\
n &= \frac{1}{\hat{\lambda}} \sum_{i=1}^{n} x_i \\
\hat{\lambda} &= \frac{1}{n} \sum_{i=1}^{n} x_i
\end{aligned}$$

Problem 3 (25 marks)

Text Data Classification - In this question, we will dry-run the Naïve Bayes' (parts (b)-(d)) classification algorithm to predict if the sentiment of the text data provided. You are provided the following reviews and labels:

Sentiment	Text	
Training	Positive	very enjoyable
	Positive	really surprising
	Positive	really fun and enjoyable
	Negative	not surprising
	Negative	very boring and predictable
Test	?	pretty predictable and no fun

Table 1: Review data

(a) [**5 marks**] First let us use the trained weights provided in Table 2 to predict the sentiment of the test data. We label a positive review as '1' and negative review as '-1', take the decision boundary to be 0. You are provided a description of the features and their respective trained weights in the table below. Use this information to predict the sentiment of the test review. (Make appropriate assumptions about positive and negative words)

θ_i	Feature Description	Weight Value
θ_0	Bias	0.6
θ_1	Count of Positive Words	1.2
θ_2	Count of Negative Words	-3.5
θ_3	log(word count)	0.1

Table 2: Features and Trained Weights

(b) [**10 marks**] We now use Naïve Bayes to predict the sentiment of the review, use 'Laplace add-1 smoothing' on Table 1 to compute the likelihoods for all words in training data. Populate the table given below:

Word	P(Word $\vert +$)	P(Word $\vert -$)
very		
enjoyable		
⋮		

Table 3: Likelihoods of Training Data

(c) [**5 marks**] Using the table in part (b), predict the sentiment of the test data.

(d) [**5 marks**] Comment on the usefulness of Nav̈e Bayes by focusing on problems associated with predicting labels without the assumption of conditional independence. You may choose to use the problem provided to you in this question to augment your statements.

Solution:

(a) We have:

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ \ln(5) \end{bmatrix}, \boldsymbol{\theta} = \begin{bmatrix} 0.6 \\ 1.2 \\ -3.5 \\ 0.1 \end{bmatrix}$$

Our prediction is given by:

$$h(\boldsymbol{\theta}, \boldsymbol{x}),$$

Which we use to find our label and therefore sentiment using the following function:

$$y = \begin{cases} 1 & h(\boldsymbol{\theta}, \boldsymbol{x}) > 0 \\ -1 & h(\boldsymbol{\theta}, \boldsymbol{x}) < 0 \end{cases}$$

Note that when $h(\boldsymbol{\theta}, \boldsymbol{x})$ lies on the decision boundary, i.e., 0, we can randomly choose any label.
We can compute our prediction as:

$$\begin{aligned} h(\boldsymbol{\theta}, \boldsymbol{x}) &= \boldsymbol{\theta} \cdot \boldsymbol{x} \\ &= 0.6 * 1 + 1.2 * 2 + (-3.5) * 1 + 0.1 * \ln(5) \\ &= -0.34 < 0 \end{aligned}$$

Therefore, we can predict our test review as 'negative'.

(b) The populated table is given below:

Word	P(Word |+)	P(Word |−)
very	2/17	2/15
enjoyable	3/17	1/15
really	3/17	1/15
surprising	2/17	2/15
fun	2/17	1/15
and	2/17	2/15
not	1/17	2/15
boring	1/17	2/15
predictable	1/17	2/15

Table 4: Likelihoods of Training Data

(c) Using Naïve Bayes assumption, we can define our probabilities as:

$$\begin{aligned} P(+|Data) &= P(pretty|+)P(predictable|+)P(and|+)P(no|+)P(fun|+)P(+) \\ P(-|Data) &= P(pretty|-)P(predictable|-)P(and|-)P(no|-)P(fun|-)P(-) \end{aligned}$$

We can get rid of the new words and rewrite as follows:

$$\begin{aligned} P(+|Data) &= P(predictable|+)P(and|+)P(fun|+)P(+) \\ P(-|Data) &= P(predictable|-)P(and|-)P(fun|-)P(-) \end{aligned}$$

We can calculate a priori probabilities as;

$$\begin{aligned} P(+) &= 3/5 \\ P(-) &= 2/5 \end{aligned}$$

Finally we can calculate probabilities as:

$$\begin{aligned}
P(+|Data) &= P(predictable|+)P(and|+)P(fun|+)P(+) \\
&= (1/17) * (2/17) * (2/17) * (3/5) \\
&= 0.000488 \\
P(-|Data) &= P(predictable|-)P(and|-)P(fun|-)P(-) \\
&= (2/15) * (2/15) * (1/15) * (2/5) \\
&= 0.000474
\end{aligned}$$

Since $P(+|Data) > P(-|Data)$, we predict the sentiment as ‘positive’.

Problem 4 (20 marks)

Multinomial Logistic Regression Formulation - In this question we will formulate an image classification problem through logistic regression. We will classify fruit images as "orange", "apple", "strawberry", "mango", or "grape". Assume that the image data provided to us is in the form of n grey-scale 1024×1024 matrices. We can define our feature matrix for the i-th image as: $\boldsymbol{x_i} \in \mathbb{R}^{1024 \times 1024}$.

(a) [**3 marks**] How will you convert $\boldsymbol{x_i}$ feature matrix to a feature vector? What will be the size of this feature vector?

(b) [**5 marks**] Define a label to index mapping for this model. Use it to convert the following ordered labels ($n = 8$) into a vector, $\mathbf{y} \in \mathbb{R}^8$.

`orange, mango, strawberry, apple, apple, grape, mango, strawberry`

(c) [**3 marks**] What will be the shape of our weights matrix Θ?

(d) [**3 marks**] Write down the shape of our prediction matrix, i.e., the product of our feature matrix (with bias) and weights matrix.

(e) [**6 marks**] Finally, you pass your prediction matrix through the activation function, softmax. How will you extract the predicted label from this resulting matrix?

Solution:

(a) Use image flattening technique, feature vector will be of length 1024*1024.

(b) Using:

$$\begin{aligned} \text{orange} &=> 0 \\ \text{apple} &=> 1 \\ \text{strawberry} &=> 2 \\ \text{mango} &=> 3 \\ \text{grape} &=> 4 \end{aligned},$$

Our label vector for this case becomes:

$$\boldsymbol{y} = \begin{bmatrix} 0 \\ 3 \\ 2 \\ 1 \\ 1 \\ 4 \\ 3 \\ 2 \end{bmatrix}$$

(c) Θ is of size $(1024 * 1024 + 1) \times 5$ (Or $5 \times (1024 * 1024 + 1)$ depending on feature matrix and matrix multiplication order).

(d) Prediction matrix is of size $n \times 5$. Also award marks if n is taken as 8. (Or $5 \times n$ depending on feature matrix and matrix multiplication order).

(e) Use argmax on every row vector to find the index with highest probability for that training instance (or equivalent answer depending on the formulation).

Problem 5 (10 marks)

Regularized Logistic Regression - For binary logistic regression, we obtain $\boldsymbol{\theta}$ by solving the following optimization problem (see lecture notes for the notation adopted here).

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} y_i \log(h_{\boldsymbol{\theta}}(\mathbf{x}_i)) + (1 - y_i)\log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i))$$

We can extend the regularization introduced for linear regression for logistic regression. For example L_2 regularized logistic regression can be formulated by adding L_2 norm of the weight vector $\boldsymbol{\theta}$ as a penalty term $\mathcal{R}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ and formulating the optimization problem as

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}\|_2^2$$

We derived the gradient descent step size in the lectures for the loss function with penalty. Here we require you to derive the gradient of the regularized loss function with respect to $\boldsymbol{\theta}$.

Solution:

$$\begin{aligned}
\frac{\partial \mathcal{L}_\lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}}(\mathcal{L}(\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}\|_2^2) \\
&= -\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\theta}}(y_i \log(h_{\boldsymbol{\theta}}(\mathbf{x}_i)) + (1 - y_i)\log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i))) + \frac{\partial}{\partial \boldsymbol{\theta}}(\lambda\|\boldsymbol{\theta}\|_2^2) \\
&= -\sum_{i=1}^{n} \left(\left(y_i \frac{1}{h_{\boldsymbol{\theta}}(\mathbf{x}_i)} - (1 - y_i)\frac{1}{1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i)} \right) \frac{\partial}{\partial \boldsymbol{\theta}}(h_{\boldsymbol{\theta}}(\mathbf{x}_i)) \right) + 2\lambda\boldsymbol{\theta}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}}(h_{\boldsymbol{\theta}}(\mathbf{x}_i)) &= \frac{e^{-\boldsymbol{\theta}^T\mathbf{x}_i}}{(1 + e^{-\boldsymbol{\theta}^T\mathbf{x}_i})^2} \frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^T\mathbf{x}_i) \\
&= \frac{e^{-\boldsymbol{\theta}^T\mathbf{x}_i}}{(1 + e^{-\boldsymbol{\theta}^T\mathbf{x}_i})} \frac{1}{(1 + e^{-\boldsymbol{\theta}^T\mathbf{x}_i})}\mathbf{x}_i \\
&= h_{\boldsymbol{\theta}}(\mathbf{x}_i)(1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i))\mathbf{x}_i
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}_\lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= -\sum_{i=1}^{n} \left(\left(y_i \frac{1}{h_{\boldsymbol{\theta}}(\mathbf{x}_i)} - (1 - y_i)\frac{1}{1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i)} \right) (h_{\boldsymbol{\theta}}(\mathbf{x}_i)(1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i))\mathbf{x}_i) \right) + 2\lambda\boldsymbol{\theta} \\
&= -\sum_{i=1}^{n} (-(h_{\boldsymbol{\theta}}(\mathbf{x}_i)) - y_i)\mathbf{x}_i) + 2\lambda\boldsymbol{\theta} \\
&= \sum_{i=1}^{n}((h_{\boldsymbol{\theta}}(\mathbf{x}_i)) - y_i)\mathbf{x}_i) + 2\lambda\boldsymbol{\theta}
\end{aligned}$$

— End of Homework —