E0 270 Machine Learning
Assignment - 1

Sruthi Gorantla
M. Tech. CSA
SR No. 15190

February 3, 2018

1. **Probability Review.**

   (a) (3 points) Suppose that X and Y are discrete random variables. Their joint *pmf* function $P(X, Y)$ is given by the table below. Show how to compute $P(X)$, $P(Y)$, $P(X \mid Y)$ and $P(Y \mid X)$ (use conditional probabilities)

		X			
		x1	x2	x3	x4
	y1	0.1	0.2	0.03	0.07
Y	y2	0.2	0.1	0.1	0.2

Solution:

Consider that p_{ij} represents the value in the cell (x_i, y_j) in the table. i.e., then, the following are the formulae for the necessary calculations:

$$P(X) = P(X = x_i) = \sum_{j=1}^{2} P(X = x_i \mid Y = y_j) = \sum_{j=1}^{2} p_{ij}$$

$$P(Y) = P(Y = y_j) = \sum_{i=1}^{4} P(Y = y_j \mid X = x_i) = \sum_{i=1}^{4} p_{ij}$$

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)} \qquad [\because Bayes'Rule]$$

$$\implies P(X = x_i \mid Y = y_j) = \frac{P(Y = y_j \mid X = x_i)P(X = x_i)}{P(Y = y_j)} = p_{ij}$$

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)} \qquad [\because Bayes'Rule]$$

$$\implies P(Y = y_j \mid X = x_i) = \frac{P(X = x_i \mid Y = y_j)P(Y = y_j)}{P(X = x_i)} = p_{ij}$$

The computations are as follows:

$$\begin{aligned}
P(X = x_1) &= P(X = x_1 \mid Y = y_1) + P(X = x_1 \mid Y = y_2) \\
&= p_{11} + p_{12} = 0.1 + 0.2 = 0.3 \\
P(X = x_2) &= P(X = x_2 \mid Y = y_1) + P(X = x_2 \mid Y = y_2) \\
&= p_{21} + p_{22} = 0.2 + 0.1 = 0.3 \\
P(X = x_3) &= P(X = x_3 \mid Y = y_1) + P(X = x_3 \mid Y = y_2) \\
&= p_{31} + p_{32} = 0.03 + 0.1 = 0.13 \\
P(X = x_4) &= P(X = x_4 \mid Y = y_1) + P(X = x_4 \mid Y = y_2) \\
&= p_{41} + p_{42} = 0.07 + 0.2 = 0.27
\end{aligned}$$

$$\begin{aligned}
P(Y = y_1) &= P(Y = y_1 \mid X = x_1) + P(Y = y_1 \mid X = x_2) \\
&\quad + P(Y = y_1 \mid X = x_3) + P(Y = y_1 \mid X = x_4) \\
&= p_{11} + p_{21} + p_{31} + p_{41} \\
&= 0.1 + 0.2 + 0.03 + 0.07 = 0.4 \\
P(Y = y_2) &= P(Y = y_2 \mid X = x_1) + P(Y = y_2 \mid X = x_2) \\
&\quad + P(Y = y_2 \mid X = x_3) + P(Y = y_2 \mid X = x_4) \\
&= p_{12} + p_{22} + p_{32} + p_{42} \\
&= 0.2 + 0.1 + 0.1 + 0.2 = 0.6
\end{aligned}$$

$$\begin{aligned}
P(X = x_1 \mid Y = y_1) &= p_{11} = 0.1 & P(X = x_1 \mid Y = y_2) &= p_{12} = 0.2 \\
P(X = x_2 \mid Y = y_1) &= p_{21} = 0.2 & P(X = x_2 \mid Y = y_2) &= p_{22} = 0.1 \\
P(X = x_3 \mid Y = y_1) &= p_{31} = 0.03 & P(X = x_3 \mid Y = y_2) &= p_{32} = 0.1 \\
P(X = x_4 \mid Y = y_1) &= p_{41} = 0.07 & P(X = x_4 \mid Y = y_2) &= p_{42} = 0.2
\end{aligned}$$

$$\begin{aligned}
P(Y = y_1 \mid X = x_1) &= p_{11} = 0.1 & P(Y = y_1 \mid X = x_2) &= p_{21} = 0.2 \\
P(Y = y_1 \mid X = x_3) &= p_{31} = 0.03 & P(Y = y_1 \mid X = x_4) &= p_{41} = 0.07 \\
P(Y = y_2 \mid X = x_1) &= p_{12} = 0.2 & P(Y = y_2 \mid X = x_2) &= p_{22} = 0.1 \\
P(Y = y_2 \mid X = x_3) &= p_{32} = 0.1 & P(Y = y_2 \mid X = x_4) &= p_{42} = 0.2
\end{aligned}$$

(b) (3 points) Let X be a normally distributed random variable with zero mean and standard deviation of 1. What is P(X = 0.6)?

Solution:

Since X is a normally distributed random variable, the probability density function is given by:

$$p(X = x) = \frac{1}{\sqrt{2\pi}\sigma} exp\Big[- \frac{1}{2}\Big(\frac{x - \mu}{\sigma}\Big)^2\Big]$$

Given mean (μ) = 0 and standard deviation (σ) = 1. Substituting μ, σ and $x = 0.6$ in the above formula, we get:

$$p(X = 0.6) = \frac{1}{\sqrt{2\pi}(1)} exp\Big[- \frac{1}{2}\Big(\frac{(0.6) - (0)}{(1)}\Big)^2\Big] = 0.33322460289$$

(c) (4 points) A bag contains 2 unbiased coins. One coin has heads and tails on opposing sides, while the other has tails on both the sides. You put your hand in the bag and take out a coin. Without looking at the coin, you flip it. When the coin lands on the floor, you observe that the side facing you reads tails. What is the probability that the opposite side (the one you cannot see) is also tails?

Solution:

Let X be the random variable which can take two values, x_1 denotes that the coin has tails on the other side too and x_2 denotes that the coin has heads on the other side. The probability that the opposite side of the coin flipped is also tails is equal to the probability that we picked the coin with tails on both sides. Among the two coins, only one such coin exists (the other one has tails on one side and heads on the other side). Hence,

$$P(X = x_1) = \frac{1}{2}$$

2. (5 points) Given a two class problem. $Y \in \{0, 1\}$, derive the posterior class probabilities, $P(Y = i \mid X)$ given Poisson distributed class conditional densities of the form: $P(X \mid Y = i) = \Pi_j \frac{\lambda_i^{x_j}}{x_j!} e^{-\lambda_i}$ and $P(Y = 0) = \theta$.

Solution:

Given are the Poisson distributed class conditional densities of the form:

$$P(X \mid Y = i) = \Pi_j \frac{\lambda_i^{x_j}}{x_j!} e^{-\lambda_i}$$
$$P(Y = 0) = \theta \implies P(Y = 1) = 1 - \theta$$

To find the posterior class probabilities, we use Bayes' Rule.

$$P(A = i \mid B) = \frac{P(B \mid A = i)P(A = i)}{P(B)} = \frac{P(B \mid A = i)P(A = i)}{\sum_{j=1}^{n} P(B \mid A = j)P(A = j)}$$

$$\begin{aligned}
P(Y = 0 \mid X) &= \frac{P(X \mid Y = 0)P(Y = 0)}{P(X \mid Y = 0)P(Y = 0) + P(X \mid Y = 1)P(Y = 1)} \\
&= \frac{(\Pi_j \frac{\lambda_0^{x_j}}{\cancel{x_j!}} e^{-\lambda_0})(\theta)}{(\Pi_j \frac{\lambda_0^{x_j}}{\cancel{x_j!}} e^{-\lambda_0})(\theta) + \Pi_j \frac{\lambda_1^{x_j}}{\cancel{x_j!}} e^{-\lambda_1}(1-\theta)} \\
&= \frac{1}{1 + \frac{(1-\theta)}{\theta} e^{\lambda_0 - \lambda_1} \Pi_j (\frac{\lambda_1}{\lambda_0})^{x_j}}
\end{aligned}$$

$$\begin{aligned}
P(Y = 1 \mid X) &= \frac{P(X \mid Y = 1)P(Y = 1)}{P(X \mid Y = 0)P(Y = 0) + P(X \mid Y = 1)P(Y = 1)} \\
&= \frac{(\Pi_j \frac{\lambda_1^{x_j}}{\cancel{x_j!}} e^{-\lambda_1})(1-\theta)}{(\Pi_j \frac{\lambda_0^{x_j}}{\cancel{x_j!}} e^{-\lambda_0})(\theta) + \Pi_j \frac{\lambda_1^{x_j}}{\cancel{x_j!}} e^{-\lambda_1}(1-\theta)} \\
&= \frac{1}{1 + \frac{\theta}{(1-\theta)} e^{\lambda_1 - \lambda_0} \Pi_j (\frac{\lambda_0}{\lambda_1})^{x_j}}
\end{aligned}$$

3. (5 points) Given a 2-class classification problem in 2 dimensional space. What is the Bayes decision boundary for this problem.
$p(\boldsymbol{x} \mid w_1) \sim N([2,1]^T, I)$
$p(\boldsymbol{x} \mid w_2) \sim N([1,1]^T, I)$
and $P(w_1) = 0.2, P(w_2) = 0.8$

Solution:

The minimum-error-rate classification can be achieved by use of the discriminant functions

$$g_i(\boldsymbol{x}) = \ln p(\boldsymbol{x} \mid w_i) + \ln P(w_i)$$

Given $p(\boldsymbol{x} \mid w_i)$ are multivariate normal distributions. Hence,

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i}) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(w_i)$$

For the given distributions,

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Hence this comes under case 1: $\boldsymbol{\Sigma}_i = \sigma^2 I$ where $\sigma = 1$. Hence we directly use the equations given [1]. The determinants are easy to find. $|\boldsymbol{\Sigma}_i| = \sigma^{2d} = 1$. We can ignore the terms $|\boldsymbol{\Sigma}_i|$ and $\frac{d}{2}\ln 2\pi$ as they are not dependent on i. Thus our discriminant functions are:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^t \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i}) + \ln P(w_i)$$

The inverse matrices are simple in this case:

$$\Sigma_1^{-1} = \Sigma_2^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Hence $g_i(\boldsymbol{x})$ can be calculated as below:

$$\begin{aligned} g_i(\boldsymbol{x}) &= -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^t(\boldsymbol{x} - \boldsymbol{\mu_i}) + \ln P(w_i) \\ &= -\frac{1}{2\sigma^2}\Big(\boldsymbol{x}^t\boldsymbol{x} - 2\boldsymbol{\mu}_i^t\boldsymbol{x} + \boldsymbol{\mu}_i^t\boldsymbol{\mu}_i\Big) + \ln P(w_i) \end{aligned}$$

Ignoring $\boldsymbol{x}^t\boldsymbol{x}$ as it is independent of i, we have a linear discriminant function:

$$g_i(\boldsymbol{x}) = -\frac{1}{2\sigma^2}\Big(- 2\boldsymbol{\mu}_i^t\boldsymbol{x} + \boldsymbol{\mu}_i^t\boldsymbol{\mu}_i\Big) + \ln P(w_i)$$

Given $\boldsymbol{\mu_1} = [2,1]^T$, $\boldsymbol{\mu_2} = [1,1]^T$ and $P(w_1) = 0.2, P(w_2) = 0.8$. Substituting these values in the above equation, we get:

$$\begin{aligned} g_1(\boldsymbol{x}) &= -\frac{1}{2}\Bigg(- 2\begin{pmatrix} 2 \\ 1 \end{pmatrix}^t \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix}^t \begin{pmatrix} 2 \\ 1 \end{pmatrix}\Bigg) + \ln 0.2 \\ &= -\frac{1}{2}\Big((-4x_1 - 2x_2) + (5) \Big) + \ln 0.2 \\ &= 2x_1 + x_2 - \frac{5}{2} + \ln 0.2 \end{aligned}$$

$$\begin{aligned} g_2(\boldsymbol{x}) &= -\frac{1}{2}\Bigg(-2\begin{pmatrix}1\\1\end{pmatrix}^t \begin{pmatrix}x_1\\x_2\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}^t \begin{pmatrix}1\\1\end{pmatrix} \Bigg) + \ln 0.8 \\ &= -\frac{1}{2}\Big(\big(-2x_1 - 2x_2\big) + \big(2\big) \Big) + \ln 0.8 \\ &= x_1 + x_2 - 1 + \ln 0.8 \end{aligned}$$

We set $g_1(\boldsymbol{x}) = g_2(\boldsymbol{x})$ to find the decision boundary:

$$\begin{aligned} 2x_1 + x_2 - \frac{5}{2} + \ln 0.2 &= x_1 + x_2 - 1 + \ln 0.8 \\ \implies x_1 &= \frac{3}{2} + 2\ln 2 \end{aligned}$$

Hence, $x_1 = \frac{3}{2} + 2\ln 2$ is the required Bayes decision boundary for the given distributions.

4. The conditional independence assumptions made by Naive Bayes may not hold in reality. Inspite of that fact, Naive Bayes can often do quite well at classification tasks. Consider an example where X_1, X_2 and X_3 are all Boolean features and Y is a Boolean label. X_1 and X_2 are truly independent given Y and X_3 is a copy of X_2 (meaning that X_3 and X_2 always have the same value). Suppose you are now given a test example with $X_1 = T$ and $X_2 = X_3 = F$. You are also given the probabilities: $P(X_1 = T \mid Y = T) = p$; $P(X_1 = T \mid Y = F) = 1 - p$; $P(X_2 = F \mid Y = T) = q$; $P(X_2 = F \mid Y = F) = 1 - q$; $P(Y = T) = P(Y = F) = 0.5$

   (a) (5 points) Prove that the Naive Bayes decision rule for classifying the test example positively is: $p \geq \frac{(1-q)^2}{q^2 + (1-q)^2}$

   Solution:

   Let $x_1 = T, x_2 = x_3 = F$. Then, Naive Bayes decision rule is given by:

   Predict $Y = T$ if:

$$P(Y = T)\prod_{i=1}^{3} P(X_i = x_i \mid Y = T) \geq P(Y = F)\prod_{i=1}^{3} P(X_i = x_i \mid Y = F)$$

$$\prod_{i=1}^{3} P(X_i = x_i \mid Y = T) \geq \prod_{i=1}^{3} P(X_i = x_i \mid Y = F)$$
$$[\because P(Y = T) = P(Y = F) = 0.5]$$

   Substituting the values of x_1, x_2, x_3, we get,

$$\begin{aligned} pq^2 &\geq (1-p)(1-q)^2 \\ \frac{p}{(1-p)} &\geq \frac{(1-q)^2}{q^2} \\ \frac{q^2}{(1-q)^2} &\geq \frac{(1-p)}{p} \\ \frac{q^2 + (1-q)^2}{(1-q)^2} &\geq \frac{1}{p} \\ \therefore p &\geq \frac{(1-q)^2}{q^2 + (1-q)^2} \end{aligned}$$

(b) (5 points) What is the true decision rule for classifying the test example positively, in terms of p and q?

Solution:

To find the true decision boundary, we have to consider that X_3 depends on X_2. Hence, we classify the given example as positive if:

$$P(Y = T \mid X_1 = T, X_2 = F, X_3 = F) \geq P(Y = F \mid X_1 = T, X_2 = F, X_3 = F)$$
$$\frac{P(X_1 = T \wedge X_2 = F \wedge X_3 = F \wedge Y = T)}{P(X_1 = T \wedge X_2 = F \wedge X_3 = F)} \geq \frac{P(X_1 = T \wedge X_2 = F \wedge X_3 = F \wedge Y = F)}{P(X_1 = T \wedge X_2 = F \wedge X_3 = F)}$$

Cancelling the denominator on both sides,

$$P(X_1 = T \wedge X_2 = F \wedge X_3 = F \wedge Y = T) \geq P(X_1 = T \wedge X_2 = F \wedge X_3 = F \wedge Y = F)$$

$$\begin{aligned} &P(Y = T)P(X_1 = T \mid Y = T)P(X_2 = F \mid Y = T)P(X_3 = F \mid X_2 = F \wedge Y = T) \\ &\quad \geq P(Y = F)P(X_1 = T \mid Y = F)P(X_2 = F \mid Y = F)P(X_3 = F \mid X_2 = F \wedge Y = F) \end{aligned}$$

$P(X_3 = F \mid X_2 = F \wedge Y = T)$ is given by

$$P(X_3 = F \mid X_2 = F \wedge Y = T) = \frac{P(X_3 = F \wedge X_2 = F \wedge Y = T)}{P(X_2 = F \wedge Y = T)}$$

But we know that $X_3 = F$ whenever $X_2 = F$. Hence, $X_3 = F \wedge X_2 = F$ is same as $X_2 = F$. Hence,

$$P(X_3 = F \mid X_2 = F \wedge Y = T) = \frac{P(X_2 = F \wedge Y = T)}{P(X_2 = F \wedge Y = T)} = 1$$

Similarly,

$$P(X_3 = F \mid X_2 = F \wedge Y = F) = \frac{P(X_2 = F \wedge Y = F)}{P(X_2 = F \wedge Y = F)} = 1$$

Therefore, classify the new example as positive if:

$$\begin{aligned} &P(Y = T)P(X_1 = T \mid Y = T)P(X_2 = F \mid Y = T) \\ &\qquad \geq P(Y = F)P(X_1 = T \mid Y = F)P(X_2 = F \mid Y = F) \end{aligned}$$

Cancelling the terms $P(Y = T)$ and $P(Y = F)$ from both sides,

$$\begin{aligned} pq &\geq (1 - p)(1 - q) \\ pq &\geq 1 - p - q + pq \\ p &\geq 1 - q \end{aligned}$$

Therefore, $p \geq 1 - q$ is the true decision boundary.

(c) (5 points) Plot the two decision boundaries (vary q between 0 and 1) and show where the first rule makes mistakes relative to the true decision rule.

Solution:

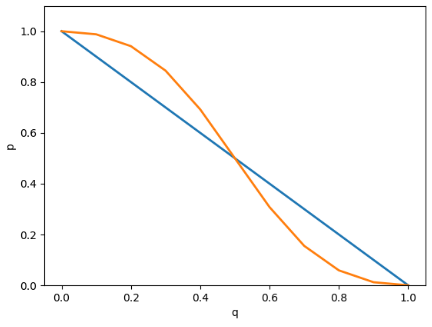

Figure 1: True and Naive Bayes Decision Boundaries

In the figure 1, the blue curve represents the true decision boundary and the red curve represents the Naive Bayes' decision boundary.

The region in between the two curves is the region where the Naive Bayes' decision rule makes mistakes relative to the true decision rule.

5. **Text Classification using Naive Bayes** In this problem, we will use the naive Bayes algorithm for text classification. The dataset for this problem is a R8 subset of the Reuters-21578 dataset, which can be obtained from this website (look at the R8 dataset). Read the website for more details about the dataset. You have been provided with separate training and test files containing 5485 and 2189 articles (samples) respectively. Each article comes from one of the eight categories (class label).

   (a) (2 points) Process the raw data and create one hot vector encoding for the test and train data. Convert the words to lowercase and sort the words lexicographically before forming the index for one hot vector encoding. Plot the word frequencies(log scale) on a graph and provide this graph. Read about zipfs law. Is the zipfs law valid in the corpus?

   Solution:

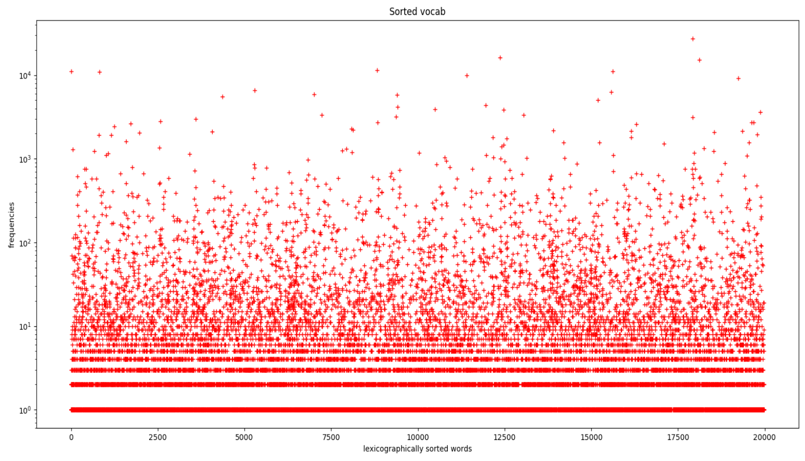

Figure 2: log frequencies of words sorted in lexicographic order

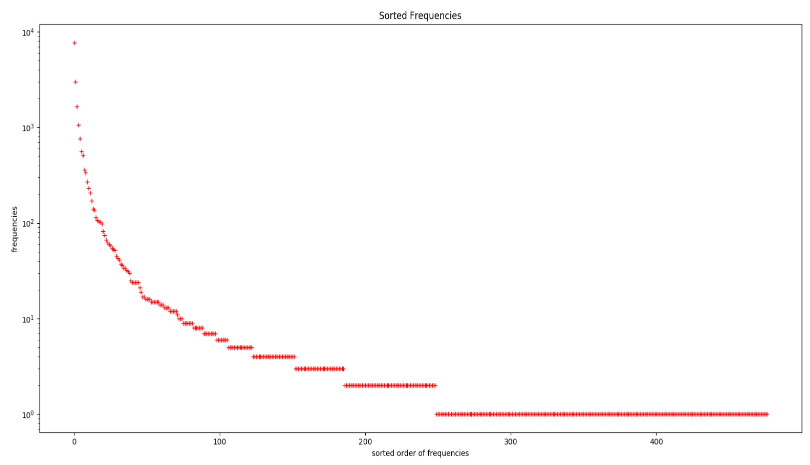

Figure 3: log frequencies of words sorted in descending order of frequencies

The first few highest frequencies in the dataset are:
7693, 2994, 1651, 1059, 759, 559, 509, 359, 335, 271, 232. ..... ,1
From here we can notice that, the first most frequent word appears almost two times as frequent as the second most frequent word and similarly almost three times as frequent as the third most frequent word. By looking at the log scale of decreasing order of frequencies in Figure 3, we can conclude that this given dataset follows **zipf's law**.

(b) (5 points) Implement the Naive Bayes algorithm to classify each of the articles into one of the given categories, using one hot vector encoding. Report the accuracies over the training as well as the test set. You should use Laplace Smoothing to avoid zero probabilities.

Solution:

Train accuracy: 0.7649954421148587
Test accuracy: 0.7775239835541343

(c) (5 points) Read about the confusion matrix. Draw the confusion matrix for your results in the part (a) above for the test data. Which category has the highest value of the diagonal entry in the confusion matrix? What does that mean? Which two categories are confused the most with each other i.e. which is the highest entry among the non-diagonal entries in the confusion matrix? Explain your observations. Include the confusion matrix in your submission.

Solution:

The table 1 shows the confusion matrix for the test data. As we can see, the class *earn* has the highest value *1062* among the diagonal entries which means it has been classified correctly compared to all other classes by the Naive Bayes' Classifier. The highest value among the non-diagonal entries is *90* which represents that the class *acq* has been incorrectly classified as the class *earn*.

Predicted Labels									
		earn	money-fx	trade	acq	grain	interest	crude	ship
True Labels	earn	1062	0	0	21	0	0	0	0
	money-fx	27	0	0	60	0	0	0	0
	trade	16	0	10	48	1	0	0	0
	acq	90	0	0	606	0	0	0	0
	grain	5	0	0	5	0	0	0	0
	interest	32	1	0	43	0	5	0	0
	crude	37	0	0	65	0	0	19	0
	ship	6	0	0	29	0	0	1	0

Table 1: Confusion Matrix

(d) (5 points) Read about stemming and stop word removal. Clean up the raw data, by applying stop word removal and Porter stemming algorithm using NLTK Python Library. Learn a new model on the cleaned up data. Again, report the accuracy as well as the confusion over the test data. How do your accuracies and confusion matrix change? Comment on your observations.

Solution:

Train accuracy: 0.8018231540565178
Test accuracy: 0.8058474189127456
The accuracy of both the training set and the test set increases significantly. From the confusion matrix, we can see that this model is not most confused between *earn* and *acq* classes like in the previous case.

Predicted Labels									
		earn	money-fx	trade	acq	grain	interest	crude	ship
True Labels	earn	1066	1	0	12	0	1	1	2
	money-fx	13	9	1	54	7	1	0	2
	trade	24	2	5	28	6	4	2	4
	acq	18	5	3	646	9	3	6	6
	grain	4	0	0	6	0	0	0	0
	interest	32	2	3	34	3	4	1	2
	crude	34	3	1	33	10	4	32	4
	ship	1	0	1	27	2	1	2	2

Table 2: Confusion Matrix after stemming and stop word removal

(e) (5 points) Read about TF-IDF. Apply TF-IDF to transform your one hot vector encoding of the samples. Learn a new model on the transformed data. Again, report the accuracy as well as the confusion matrix over the test data. How do your accuracies and confusion matrix change? Comment on your observations.

Solution:

Logistic Regression

The data has been normalised using the maximum and minimum values of features in the training set:

$$x_i = \frac{x_i - min_i}{max_i - min_i}$$

Train accuracy: 0.851230628988
Test accuracy: 0.497030607583

		Predicted Labels							
		earn	money-fx	trade	acq	grain	interest	crude	ship
True Labels	earn	1080	0	0	3	0	0	0	0
	money-fx	87	0	0	0	0	0	0	0
	trade	72	0	0	3	0	0	0	0
	acq	688	0	0	8	0	0	0	0
	grain	10	0	0	0	0	0	0	0
	interest	80	0	0	1	0	0	0	0
	crude	117	0	0	4	0	0	0	0
	ship	36	0	0	0	0	0	0	0

Table 3: Confusion Matrix with TF-IDF

Naive Bayes
Train accuracy: 0.9232452142206017
Test accuracy: 0.4582000913659205
The accuracy of the tf-idf model on the train set increases while the accuracy on the test data decreases drastically compared to model with only tf. From the confusion matrix, we can see that the model is again most confused between *earn* and *acq* classes.

		Predicted Labels							
		earn	money-fx	trade	acq	grain	interest	crude	ship
True Labels	earn	600	0	0	465	2	1	13	2
	money-fx	35	0	1	44	2	1	4	0
	trade	18	0	0	48	4	1	3	1
	acq	208	8	9	395	21	3	51	1
	grain	6	0	1	3	0	0	0	0
	interest	23	0	0	55	1	1	1	0
	crude	56	1	0	54	1	1	7	1
	ship	9	0	1	26	0	0	0	0

Table 4: Confusion Matrix with TF-IDF

(f) (3 points) Please go through the data and propose any other processing steps that you can apply on the data, that you think might improve the results. Please explain why you think the improvement will happen. Implement your processing steps, learn a new model on the transformed data, report the new accuracy and confusion matrix over the test data.

Solution:

Stemming chops off the affixes of a word to reduce it from a derivationally related forms of a word to a common base form. This might sometimes lead to unwanted results where the resultant word doesn't even form a meaningful word. Hence, we use ***Lemmatization***. In computational linguistics, lemmatisation is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document [2].
We perform lemmatization to the given data and the train and test accuracies are as shown below:
Train Accuracy: 0.80820419325433
Test Accuracy: 0.8126998629511192
As we can see, there is significant improvement in the test accuracy as expected because the lemmatizer preserves the meaning of the words helping our classifier to learn better about the entire document.

Predicted Labels									
		earn	money-fx	trade	acq	grain	interest	crude	ship
True Labels	earn	1064	0	0	13	2	2	2	0
	money-fx	17	17	1	45	5	0	0	2
	trade	19	3	10	23	6	5	4	5
	acq	18	7	3	642	9	7	4	6
	grain	4	0	0	5	1	0	0	0
	interest	34	2	3	32	3	5	2	0
	crude	29	4	1	32	10	1	38	6
	ship	5	0	1	24	1	0	3	2

Table 5: Confusion Matrix with Lemmatization

6. (10 points) Derive the gradient descent update rule for L2 regularized variant of Logistic Regression objective formulated in class.

Solution:

Let D be the dataset given with n-dimensional feature vectors X_l where $1 \leq l \leq |D|$ and true labels Y_l where $Y_l \in \{0, 1\}$. In Logistic Regression, we learn $P(Y \mid X)$ directly. We assume that these probabilities take the form of a sigmoid function. Hence,

$$P(Y = 0 \mid X) = \frac{1}{1 + exp(w_0 + \Sigma_{i=1}^{n} w_i X_i)} \tag{1}$$

$$P(Y = 1 \mid X) = \frac{exp(w_0 + \Sigma_{i=1}^{n} w_i X_i)}{1 + exp(w_0 + \Sigma_{i=1}^{n} w_i X_i)} \tag{2}$$

We aim at finding optimal $\hat{W}$ such that this probability is maximized.

$$\hat{W} = \underset{W}{\text{argmax}} \prod_{l=1}^{|D|} P(Y^l \mid X^l, W) \tag{3}$$

where $W = \langle w_0, w_1, \ldots, w_n \rangle$. We can apply log on both sides since its a monotonic function. We then get,

$$l(W) = \sum_{l=1}^{|D|} \log P(Y^l \mid X^l, W) \tag{4}$$

$$l(W) = \sum_{l=1}^{|D|} Y^l \log P(Y^l = 1 \mid X^l, W) + (1 - Y^l) \log P(Y^l = 0 \mid X^l, W) \tag{5}$$

Substituting (1) and (2) in (5) we get,

$$l(W) = \sum_{l=1}^{|D|} Y^l((w_0 + \Sigma_{i=1}^{n} w_i X_i^l) - \log(1 + exp(w_0 + \Sigma_{i=1}^{n} w_i X_i^l))) + \tag{6}$$
$$(1 - Y^l)(-\log(1 + exp(w_0 + \Sigma_{i=1}^{n} w_i X_i^l)))$$

Expanding the equation (6):

$$\begin{aligned} l(W) &= \sum_{l=1}^{|D|} Y^l (w_0 + \Sigma_{i=1}^n w_i X_i^l) - Y^l \log\left(1 + exp(w_0 + \Sigma_{i=1}^n w_i X_i^l)\right) \quad (7)\\ &\quad - \log\left(1 + exp(w_0 + \Sigma_{i=1}^n w_i X_i^l)\right) + Y^l \log\left(1 + exp(w_0 + \Sigma_{i=1}^n w_i X_i^l)\right) \\ l(W) &= \sum_{l=1}^{|D|} Y^l (w_0 + \Sigma_{i=1}^n w_i X_i^l) - \log\left(1 + exp(w_0 + \Sigma_{i=1}^n w_i X_i^l)\right) \quad (8) \end{aligned}$$

To apply gradient descent update rule, we need to the find the derivative of equation (8) with respect to each w_i:

$$\begin{aligned} \frac{\partial l(W)}{\partial w_i} &= \sum_{l=1}^{|D|} X_i^l Y^l - \frac{exp(w_0 + \Sigma_{i=1}^n w_i X_i^l) X_i^l}{1 + exp(w_0 + \Sigma_{i=1}^n w_i X_i^l)} \quad (9)\\ &= \sum_{l=1}^{|D|} X_i^l (Y^l - \hat{P}(Y^l = 1 \mid X^l, W)) \quad (10)\\ \therefore w_i &= w_i - \eta \sum_{l=1}^{|D|} X_i^l (Y^l - \hat{P}(Y^l = 1 \mid X^l, W)) \quad (11) \end{aligned}$$

Hence, we can update the weights according the the equation (11). η is the learning rate.

L2 Regularization:
With L2 Regularization, our objective function has an extra term,

$$\hat{W} = \underset{W}{\text{argmax}} \prod_{l=1}^{|D|} P(Y^l \mid X^l, W) + \frac{\lambda}{2} \|W\|^2 \quad (12)$$

Then we update the weights as shown below:

$$\begin{aligned} \frac{\partial l(W)}{\partial w_i} &= \sum_{l=1}^{|D|} X_i^l (Y^l - \hat{P}(Y^l = 1 \mid X^l, W)) + \lambda w_i \quad (13)\\ \therefore w_i &= w_i - \eta \sum_{l=1}^{|D|} X_i^l (Y^l - \hat{P}(Y^l = 1 \mid X^l, W)) - \eta \lambda w_i \quad (14) \end{aligned}$$

Thus, equation (14) gives the gradient descent update rule for L2 regularized variant of Logistic Regression objective function.

7. **Text Classification using Logistic Regression** In this problem, we will use Logistic Regression for text classification. For this problem, you are provided a spam classification data set, where each instance is an email message represented by 57 features and is associated with a binary label indicating whether the email is spam (+1) or non-spam (−1). Split the data randomly to test and train set, using 80-20 split.
   (a) (5 points) Provide your training data set and test data set as a text file. If this file is not submitted, rest of this question will not be evaluated.

Solution:

Files submitted (train_data.txt, test_data.txt, train_labels.txt, test_labels.txt)
NOTE: The feature vectors have been normalized using standardisation as below:

$$x' = \frac{x - \mu}{\sigma}$$

This normalization was done to ensure that the 55th, 56th and 57th features, which have broad range of values, will not govern the classifier.

(b) (10 points) Write a piece of code to implement logistic regression on a given binary classification data set $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ is an $m \times d$ matrix (m instances, each of dimension d) and y is an m-dimensional vector (with $y_i \in \{\pm 1\}$ being a binary label associated with the i-th instance in X): your program should take the training data $(\mathbf{X}, \mathbf{y})$ as input and output a d-dimensional weight vector $\mathbf{w}$ and bias (threshold) term b representing the learnt classifier. Report the accuracies over the training as well as the test set.

Solution:

Train Accuracy: 0.933967391304
Test Accuracy: 0.942453854506
Note: The labels for both train and test data are given as $y_i \in \{1, 0\}$ in the dataset. Hence, the same labels have been used.

(c) (15 points) Use your implementation of logistic regression to learn a classifier from 10% of the training data, then 20% of the training data, then 30% and so on upto 100%. In each case, measure the classification error on the training examples used, as well as the error on the given test set. Plot a curve showing both the training error and the test error (on the y-axis) as a function of the number of training examples used (on the x-axis). Such a plot is often called a learning curve. (Note: the training error should be calculated only on the subset of examples used for training, not on all the training examples available in the given data set.)

Solution:

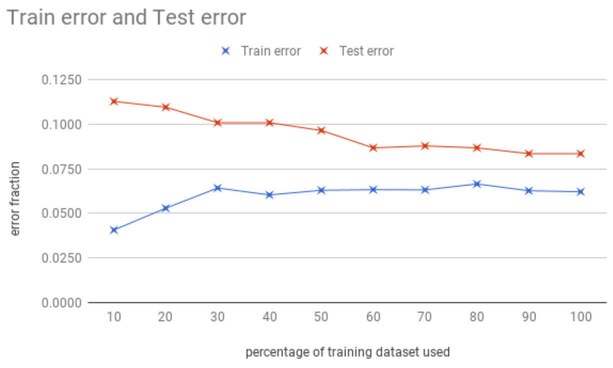

Figure 4: Learning Curve

References

[1] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Ed)*. Wiley, 2001.

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.