Assignment 1 NLP

In collaboration with Tongyan Wang

1 Text Classification - Eisenstein 4.6 (p. 89)

1.1 Sentiment lexicon-based classifier

The algorithm for the sentiment lexicon-based classifier is found under the analyze_sentiment function. The steps for the algorithm are described below:

1. From the 400 reviews hold out test set, each file is parsed and tokenized
2. Cleaning is done by removal of punctuation and stopwords
3. A full list of words is created
4. Lexicon files are parsed and 1 negative list and 1 positive list are created
5. Number of positive and negative words are determined for each file, if positive > negative words, review file would be associated to be positive, otherwise negative

Accuracy: 65.50 F1 Score: 81.40

1.2 Logistic Regression Classifier

The algorithm for the binary logistic regression classifier is found under the binary_logistic_regression function. The steps for the algorithm are described below:

1. A bag of words feature set is created for the training and test dataset. The number of features in the feature set is determined using the vocabulary words in the training and test dataset files and the feature here is whether or not the word is present.
2. The feature vector for training and test inputs will have 1600 rows for the training dataset and 400 for the test dataset and the number of columns is determined by the unique words found in the training dataset
3. The model is trained using stochastic gradient descent
4. The gradient descent outputs the weights associated with the features

Accuracy: 77.50 F1 Score: 87.32

1.3 Breaking Good

- your lexicon classifier predicts it as positive, whereas your logistic regression classifier predicts it as negative

"Though I'm an Android user, I hate to say that I like the brand new iPhone 13"

Since there are more positive words than negative words, the lexicon classifier will consider it as positive. However, the strong word "hate" is used which has a high value for the logistic regression classifier and overpowers the positive sentiments of brand new, therefore the logistic regression classifier would predict it as negative

- your lexicon classifier predicts it as negative, whereas your logistic regression classifier predicts it as positive

"this house is not that small and old as Helen described, in fact, it is really a good deal"
Since there are more negative terms than positive terms, the lexicon classifier predicts it as negative. However, the word words add a high value to the positive term good and as such, the logistic regression classifier would predict it as positive.

- Both of your classifiers predict as negative

"I cannot not take action for what you asked for."
The double negative words "cannot not" would cause both classifiers to interpret as negative sentiment however a double negative would imply a positive in human language.

1.4 Statistical Significance

$H_0: p1 = p2$

$H_A: p1 \neq p2$

$$z = \frac{\widehat{p1} - \widehat{p2}}{\sqrt{\widehat{p}(1-\widehat{p})\left(\frac{1}{n1} + \frac{1}{n2}\right)}} \text{ where } \widehat{p} = \frac{n1 \cdot \widehat{p1} + n2 \cdot \widehat{p2}}{n1 + n2}$$

$\widehat{p1} = 0.6550$ (*Accuracy of lexicon based classifier*)

$\widehat{p2} = 0.7750$ (*Accuracy of binary* log*istic regression classifier*)

$n1 = n2 = 400$

$$\widehat{p} = \frac{400 \cdot (0.6550 - 0.7750)}{800} = -0.06$$

$$\text{At } 95\% \text{ confidence interval, two-tailed } z \text{ test } \frac{z_\alpha}{2} \text{ returns } 1.96$$

$$\text{since } |z| < \frac{z_\alpha}{2}, \text{ we fail to reject the null hypothesis and the two accuracy values are statistically insignificant}$$

2. Regularization - Eisenstein 2.5 (p.44)

$$Assume\ for\ any\ feature\ j,$$
$$\theta_j^* \leq min\left(\theta_j^{(1)}, \theta_j^{(2)}\right)\ and\ \theta_j^* \geq max\left(\theta_j^{(1)}, \theta_j^{(2)}\right)$$
$$i.e.\ \exists\ \theta_j^* \in \left(\theta_j^{(1)} \cup \theta_j^{(2)}\right)^C$$
$$However,\ \theta^* \in \left(\theta^{(1)} \cup \theta^{(2)}\right)\ from\ the\ conditions\ given\ in\ the\ question,$$
$$Therefore\ by\ proof\ of\ contradiction,\ we\ show\ that:$$
$$min\left(\theta_j^{(1)}, \theta_j^{(2)}\right) \leq \theta_j^* \leq max\left(\theta_j^{(1)}, \theta_j^{(2)}\right)$$

3. XOR - Eisenstein 3.4 (p.65)

The input and desired output of the neural network are as follows:

x1	x2	f(x1, x2)
1	1	-1
1	0	1
0	1	1
0	0	-1

With 1 hidden layer, the model can be represented as follows:

$$f(\vec{x}; \vec{W}, \vec{c}, \vec{w}, \vec{b}) = sign(\vec{w}^T \cdot\ max(0,\ \vec{W}^T \cdot x + c) + b)$$

The weights are as follows:

$$W = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},\ c = \begin{pmatrix} 0 \\ -1 \end{pmatrix},\ w = \begin{pmatrix} 1 \\ 2 \end{pmatrix},\ b = 0$$

We can prove the weights as follows:

$$X \cdot W = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{pmatrix}$$

$$max(X \cdot W + c) = max\left(\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \begin{pmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{pmatrix}\right) = max\begin{pmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix}$$

$$sign((X \cdot W + c) \cdot w) = sign\left(\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -2 \end{pmatrix}\right) = sign\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

4. Initialization at Zero - Eisenstein 3.5 (pg 65)

The gradient with respect to the loss function will be the same for every weight in the weight matrix since the weights are intialized to zero. This would suggest that all the weights in the subsequent interactions will have the same value and will not change. Therefore, the model will not be able to learn the desired function from this initialization.