CSC411 project3

Xiaoxue Xing

March 25, 2017

1 Part1

This dataset has 1000 negative reviews and 1000 positive reviews. Each review has approximately 300 words. I checked the word "good", "excellent" and "terrible". Good appears 596 times in the positive reviews and 586 times in the negative reviews. Since good has the almost the same frequency in both negative and positive review, so good can't used to determine the review is positive or not. However, the excellent has 110 in positive and 35 in negative, it is useful to determine the positive review. The word terrible appears 28 times in the positive reviews and 94 in the negative reviews. So the word terrible can determine the review is negative.

2 Part2

We use the Naive Bayes. According the formula Pr(A|B) = Pr(B|A)*Pr(A)/Pr(B). So, for this question, it is Pr(class|x1,x2,...,xn) = Pr(x1,x2,...,xn|class)*Pr(class)/Pr(x1,x2,...,xn). To determine if the review is pos or neg, we just need to compare Pr(pos|x1,x2,...,xn) and Pr(neg|x1,x2,...,xn). If Pr(pos|x1,x2,...,xn) is larger, then it is positive review, if Pr(neg|x1,x2,...,xn), it is negative review.
Pr(x1,x2,...,xn|class)*Pr(class) = Pr(x1|class)*Pr(x2|class)*Pr(x3|class)...*Pr(xn|class)*Pr(class). Because Pr(xi|class) is really small, so we use exp(log(Pr(x1|class))+log(Pr(x2|class))+...+ log(Pr(xn|class)) + log(Pr(class))) I set the training set is 600, validation is 200 and test is 200. THe performance is: Training 80%, Validation 71%, and Test 69%. I set the m = 0.2 and k = 200.I use a for loop to change m and choose the best performance m.

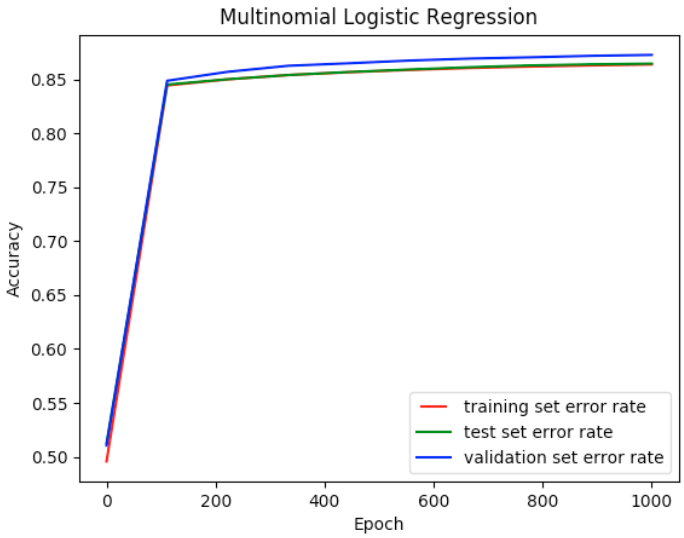
3 Part3

For the positive, the top ten I get is ['virgins', 'vh', 'undressing', 'unbuttoning', 'transylvanians', 'traditions', 'toucha', 'stocking', 'snapped', 'singable']. And the negative top 10 is ['zaltar', 'workout', 'vaccaro', 'unmercifully', 'tylenol',

'szwarc', 'stuffing', 'storyboards', 'slunk', 'scratches']. I rank the word positive dictionary and negative negative dictionary by its value, and select the first 10 keys.

4 Part4

Firstly, we construct a dataset that has all the words appear in the train set, the size of tis dataset is k. We create a k dimension vector that v[k]=1 if that word(the kth word in the dataset) appear in the review, otherwise it is zero. The output is one hot coding.Then we use the multinomial logistic regression to train it, which is similar to project2. We then use the train set to train and get the performance of train, validation and test. The model has 1 layer and the activation function is sigmoid. Iteration is 800 times. The performance is train 0.97, test 0.87,valid 0.85



5 Part5

For Naive Bayes: x is the input review. The k is the all word in the review dataset. Ii(x) indicate the ith word is in the x or not. If not in x, it will be 0, otherwise 1. θ is the log(Pr(ith word | pos))-log(Pr(ith word | neg))

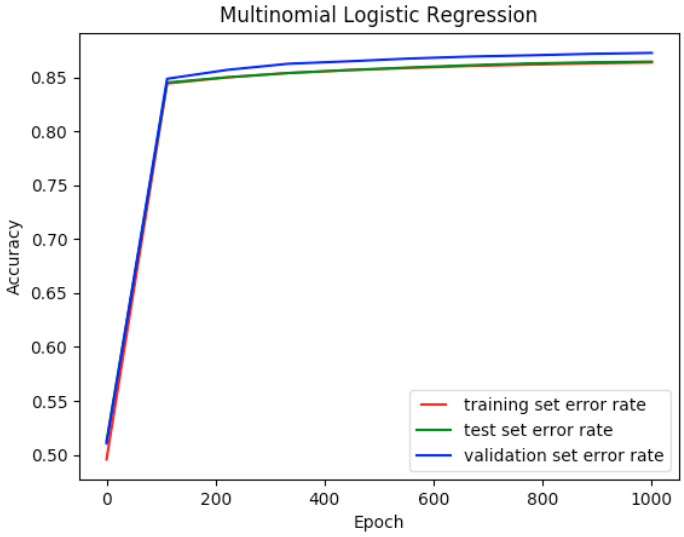
For logistic Regression: Logistic Regression is similar to Naive Bayes. The k is still the all word in the review dataset. Ii(x) indicate the ith word is in the x or not. However, this time, the θ is different. θ is the (log(Pr(ith word | pos)), log(Pr(ith word | neg)))

6 Part6

The NB average theta 0.176 and LR is 0.004. Comparing the top 100 theta, we have some common words like terrific, outstanding for good and boring, bad, stupid for bad.

7 Part7

The input layer has size of 256 and the output layer has two output. We use the same training function as part 4 but different construction method. The activation function is still sigmoid and the cost function is negative log loss function. Almost same as the part 4. The performance is train 0.86, test 0.86 and valid 0.87. The performance is and the learning curve is.



8 Part8

The 10 words for story is plot, film, benito, simmer, sitter, lift, domineering, ricci, interviews, acclaim. for good is bad, great, wonderful, reinforcing, decent, funny, manipulate, underused, admiral, perplexing