Naive Bayes and Logistic Regression for Text Classification

Hao Xiong
hxx160130@utdallas.edu

Part I Naïve Bayes

For naïve Bayes method there is not too much to talk about, one just implement the pseudo code provided in the given literature. One thing to mention is that in my java code I used a dictionary data structure, the keys storing the vocabulary and the values are arrays storing the corresponding frequency of spam and ham class. Note that the algorithm uses add-one Laplace smoothing, and I've done all the calculations in log-scale to avoid underflow.

Below is the accuracy report of Naïve Bayes method on the test set:

```
Accuracy of Naive Bayes: 0.9476987447698745
Time consumption(s): 0.5547813930000001
```

Part II Logistic Regression

In the code I've implemented both the batch gradient and incremental gradient decent (a.k.a. Stochastic gradient decent). Since the batch gradient takes much longer time than the incremental gradient when calculating the result, here I just give the accuracy of later one for this report. Let try 3 combinations of learning rate and regulation coefficient lambda:

```
learningRate = 0.01, lambda = 0.35, iteration = 1
Accuracy of Logistic Regression: 0.9560669456066946
Time consumption(s): 2.477027433
```

```
learningRate = 0.013, lambda = 0.38, iteration = 2
Accuracy of Logistic Regression: 0.897489539748954
Time consumption(s): 2.4704806660000003
```

```
learningRate = 0.009, lambda = 0.36, iteration = 10
Accuracy of Logistic Regression: 0.9518828451882845
Time consumption(s): 2.43376219
```

Part III Remove Stop Words

I just referenced the 'Default English stop words list' found at http://www.ranks.nl/stopwords when throwing the stop words. After throwing running the program again with the same parameters as above, we got the result like below:

```
Accuracy of Naive Bayes without S.W.: 0.9435146443514645
----------------------------------------------------------------------
learningRate = 0.01, lambda = 0.35, iteration = 1
Accuracy of Logistic Regression without S.W.: 0.9581589958158996
Time consumption(s): 2.477027433
----------------------------------------------------------------------
learningRate = 0.013, lambda = 0.38, iteration = 2
Accuracy of Logistic Regression without S.W.: 0.9037656903765691
----------------------------------------------------------------------
learningRate = 0.009, lambda = 0.36, iteration = 10
Accuracy of Logistic Regression without S.W.: 0.9518828451882845
```

Conclusion: For Naïve Bayes, removing stop words decreases the accuracy slightly (from 94.77% to 94.35%) while for logistic regression it helps the accuracies increased a little bit. The throwing of stop words is one of the ways to improve the feature qualities. Since most of the stop words help little when classify text, removing those low quality features results in the improvement of the accuracy does make sense. For the accuracy decreasing in Naïve Bayes I think it's because Naïve Bayes employed a Laplace smoothing which virtually assume that any word in one class at least appear once in the other class. It seems that the removing of stop words does degenerates this kind of prior assumption or somewhat.

Part IV Implement a Smoothing Method (Extra Credit)

Since in the Naïve Bayes algorithm, we've already implemented the Laplace smoothing method, here we just need to implement another smoothing method for the logistic regression. I want to let the learning rate decreases a little bit every iteration. Since large learning rate at first could make the coverage faster and little learning rate at last stage makes it possible that it could approach the coverage point even closer at last. I just added two lines of code at the end of the iteration:

```
while (repeat-- > 0) {
    for (TextVector tv : vectors) {
        for (int i = 0; i < size; i++) {
            double derivative = 0;
            if (tv.features[i] != 0)
                derivative = tv.features[i] * tv.predictionError(W);
            W[i] += learningRate * (derivative - lambda * W[i]);
        }
    }
    learningRate *= 0.95;//implement smoothing
    lambda *= 1 / 0.95;
}
```

And here is the result after smoothing.

```
learningRate = 0.009, lambda = 0.36, iteration = 10

Accuracy of Logistic Regression without S.W.: 0.9539748953974896
```

We can see that for the situation of removing stop words, after applying this smoothing method we improved the accuracy from 95.188% to 95.397%, although the improvement is not that obvious, the most important part is it works. However sometime when iterating too many times, because of overfitting the test result will not be as good as before.