COM6513, Lab4

Registration Number 170224545

08/03/2018

1. Description

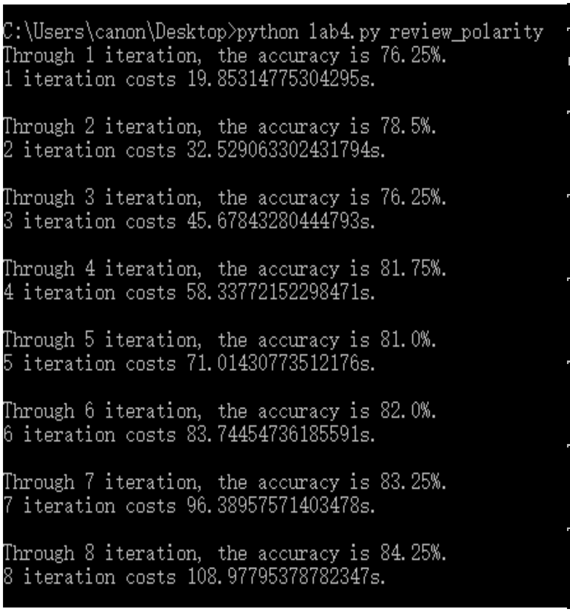
In this lab, the program successfully implements standard binary perceptron with bag-of-words representation.

This program takes the average of all the weight vectors for each class.

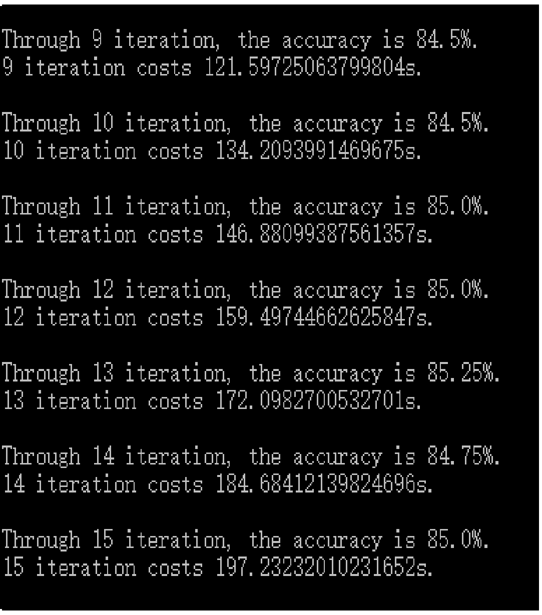
The program can be executable by running:

python3 lab4.py review_polarity

And the result is as follows:



```
C:\Users\canon\Desktop>python lab4.py review_polarity
Through 1 iteration, the accuracy is 76.25%.
1 iteration costs 19.85314775304295s.

Through 2 iteration, the accuracy is 78.5%.
2 iteration costs 32.529063302431794s.

Through 3 iteration, the accuracy is 76.25%.
3 iteration costs 45.67843280444793s.

Through 4 iteration, the accuracy is 81.75%.
4 iteration costs 58.33772152298471s.

Through 5 iteration, the accuracy is 81.0%.
5 iteration costs 71.01430773512176s.

Through 6 iteration, the accuracy is 82.0%.
6 iteration costs 83.74454736185591s.

Through 7 iteration, the accuracy is 83.25%.
7 iteration costs 96.38957571403478s.

Through 8 iteration, the accuracy is 84.25%.
8 iteration costs 108.97795378782347s.
```



```
Through 9 iteration, the accuracy is 84.5%.
9 iteration costs 121.59725063799804s.

Through 10 iteration, the accuracy is 84.5%.
10 iteration costs 134.2093991469675s.

Through 11 iteration, the accuracy is 85.0%.
11 iteration costs 146.88099387561357s.

Through 12 iteration, the accuracy is 85.0%.
12 iteration costs 159.49744662625847s.

Through 13 iteration, the accuracy is 85.25%.
13 iteration costs 172.0982700532701s.

Through 14 iteration, the accuracy is 84.75%.
14 iteration costs 184.68412139824696s.

Through 15 iteration, the accuracy is 85.0%.
15 iteration costs 197.23232010231652s.
```

It shows that the accuracies through 15 iterations is : 76.25%, 78.5%, 76.25%, 81.75%, 81.0%, 82.0%, 83.25%, 84.25%, 84.5%, 84.5%, 85.0%, 85.0%, 85.25%, 84.75%, 85.0%.

The all 15 iteration costs 197.23s.

2. Evaluation

(1) the top 10 features:

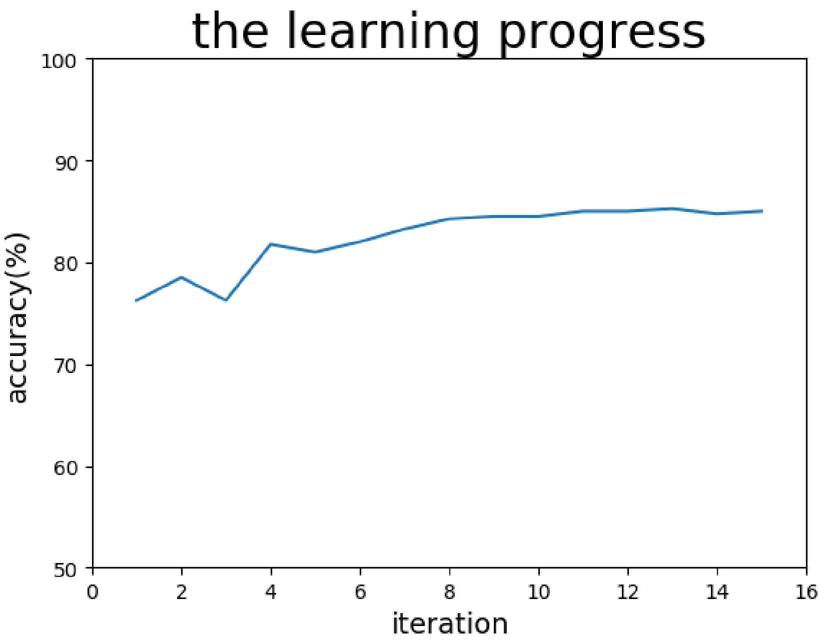
The top 10 positive features:

['seen', 'great', 'most', 'life', 'best', 'many', 'very', 'well', 'see', 'will']

The top 10 negative features:

['bad', 'script', 'only', 'plot', 'worst', 'boring', 'unfortunately', 'could', 'then', 'any']

(2) the learning progress



As we can know, the accuracy increases with the iterations. When the iteration is sufficient, the accuracy converges to 85%.

(3) Discussion of result

I think the features would generalize well for different domain because these words have obvious emotional tendencies. I propose words like “love”, “hate” and “favorite” which can show emotion can be good too.