

Report of two algorithms.

Naïve Bayes

AccuracyRate:

- Before removing Stop Words:
AccuracyRate = 0.93132183908
- After removing Stop Words:
AccuracyRate = 0.928912466844

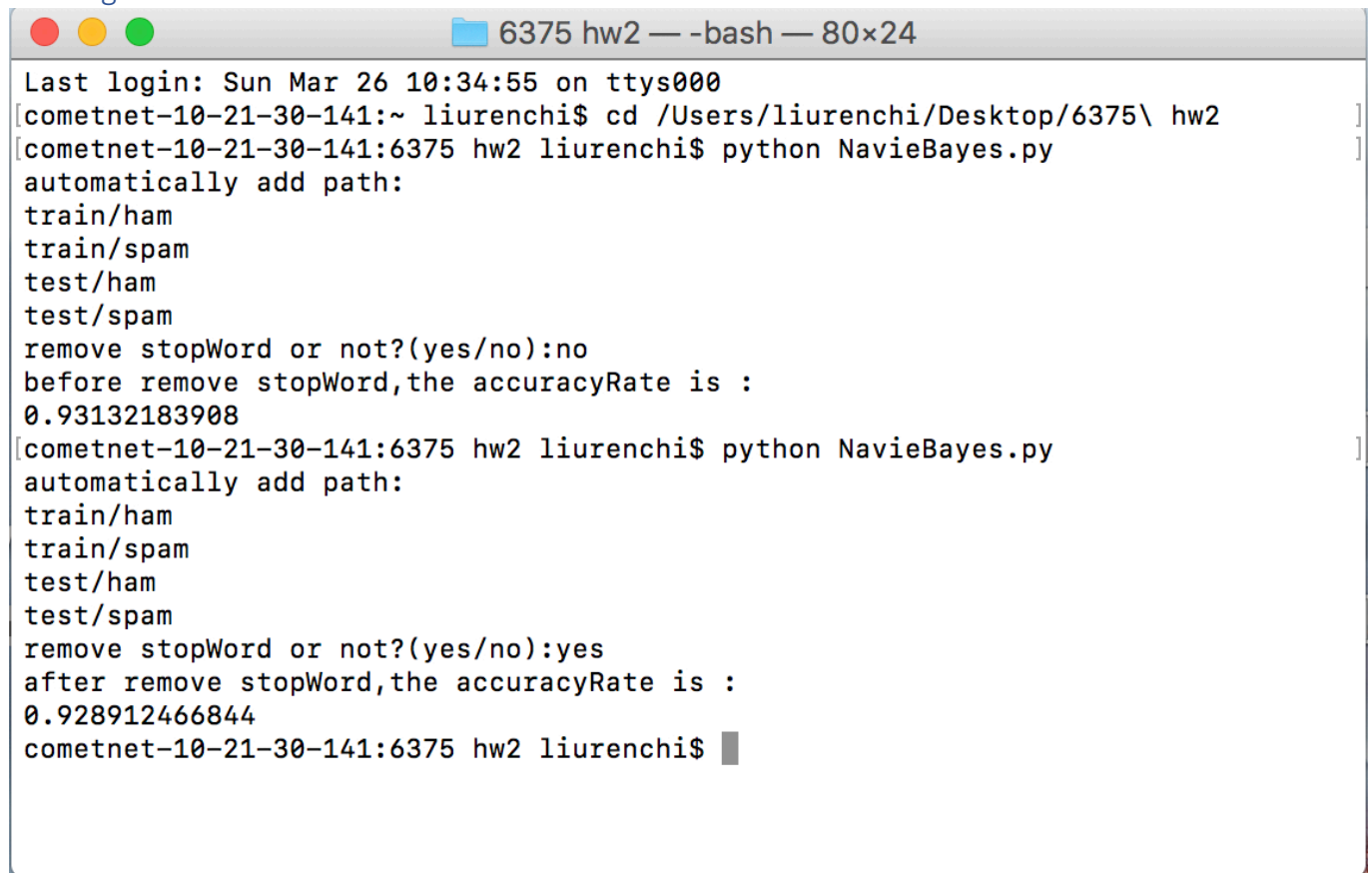
*running demo in the last part of report

Analysis:

1. After remove the stop words ,the accuracy rate decrease a little bit .The reason of this decrease may due to the difference of the Stop words's distribution in ham and spam.

By looking the stopwords and some of the email. I found this is no such big relation between email and stopword. Because some of the email only has one sentence and some of they has more stop words . This might influence the accuracy rate.

running demo:



```
6375 hw2 — -bash — 80x24
Last login: Sun Mar 26 10:34:55 on ttys000
[cometnet-10-21-30-141:~ liurenchi$ cd /Users/liurenchi/Desktop/6375\ hw2
[cometnet-10-21-30-141:6375 hw2 liurenchi$ python NavieBayes.py
automatically add path:
train/ham
train/spam
test/ham
test/spam
remove stopWord or not?(yes/no):no
before remove stopWord,the accuracyRate is :
0.93132183908
[cometnet-10-21-30-141:6375 hw2 liurenchi$ python NavieBayes.py
automatically add path:
train/ham
train/spam
test/ham
test/spam
remove stopWord or not?(yes/no):yes
after remove stopWord,the accuracyRate is :
0.928912466844
cometnet-10-21-30-141:6375 hw2 liurenchi$
```

Logistic Regression

write a test main function . in order to input data conveniently.

*code in the last part of report

learning_rate = 0.01

lambda_value \ iterations	0.01		0.1		0.5	
	before remove stopword	after remove stopword	before remove stopword	after remove stopword	before remove stopword	after remove stopword
50	0.7380	0.8135	0.7418	0.8173	0.7572	0.8212
100	0.7726	0.8289	0.7726	0.8289	0.7789	0.8366
500	0.7789	0.8481	0.7947	0.8558	0.8995	0.8933

learning_rate = 0.025

lambda_value \ iterations	0.01		0.1		0.5	
	before remove stopword	after remove stopword	before remove stopword	after remove stopword	before remove stopword	after remove stopword
50	0.7712	[0.8327	0.7712	0.8327	0.7866	0.8442
100	0.7789	0.8404	0.7789	0.8404	0.8317	0.8635
500	0.7947	0.8519	0.8519	0.8827	0.9279	0.9164

learning_rate = 0.05

lambda_value \ iterations	0.01		0.1		0.5	
	before remove stopword	after remove stopword	before remove stopword	after remove stopword	before remove stopword	after remove stopword
50	0.8087	0.8327	0.8087	0.8366	0.8510	0.8596
100	0.8087	0.8366	0.8240	0.8519	0.9058	0.9048
500	0.8317	0.8635	0.9149	0.9072	0.5461	0.9164

Analysis:

1. After remove the stopWord, the accuracy rate increase in most situations, except the 500 iterations and lambda is 0.5. this set decrease a little bit . this may because the regularization rate is too large.
2. From the data in 3 tables, I can found the overall best learning rate is 0.05.but when lambda is 0.5 and iterations the result is unstable. It may converge to a wrong point.
3. From the test result , when learning_rate = 0.05, lambda_value = 0.1, iterations = 500 or learning_rate = 0.025, lambda_value = 0.5, iterations = 500 . The logic function works best.
4. When running the program, the learning_rate is more big ,the speed of the converge is faster.

testing code: (also in logisticR.py)

```
194 # test main
195 if __name__ == "__main__":
196     print 'automatically add path:'
197     pathSet = ['train/ham','train/spam','test/ham','test/spam']
198     for p in pathSet:
199         print p
200     ham_path = pathSet[0]
201     spam_path = pathSet[1]
202     filePath1 = pathSet[2]
203     filePath2 = pathSet[3]
204     remove_stop = ['yes','no']
205     learning_rate = 0.05
206     running_time = [50,100,500]
207     lambda_value = [0.01,0.1,0.5]
208
209     result = []
210     path = [ham_path,spam_path]
211     for time in running_time:
212         resultlime = []
213         for lam in lambda_value:
214
215             for stop in remove_stop:
216                 logic = logisticRegression()
217                 StopWords = logic.readFile('stopword.txt')
218                 logic.remove_stop = stop
219                 logic.train_LR(path,learning_rate,time,lam)
220                 accuracyRate = (logic.testLRAccuracy(filePath1,'ham')+logic.testLRAccuracy(filePath2,'spam'))/2
221                 resultlime.append(accuracyRate)
222
223             result.append(resultlime)
224
225     for item in result:
226         print item
227
228     """
```

I create some parameter lists , in order to input data conveniently. With this test program,I can get one table's data.

By change the learning rate, I can get different kinds of table depend on the learning rate.

running demo:

```
6375 hw2 — -bash — 80×41
Last login: Sun Mar 26 10:34:46 on ttys000
[cometnet-10-21-30-141:~ liurenchi$ cd /Users/liurenchi/Desktop/6375\ hw2
[cometnet-10-21-30-141:6375 hw2 liurenchi$ python logisticR.py
automatically add path:
train/ham
train/spam
test/ham
test/spam
[0.7380415561450044, 0.813527851458886, 0.7418877099911583, 0.8173740053050398,
0.7572723253757736, 0.8212201591511936]
[0.7726569407603889, 0.8289124668435013, 0.7726569407603889, 0.8289124668435013
0.7789124668435013, 0.8366047745358091]
[0.7789124668435013, 0.8481432360742706, 0.794761273209549, 0.8558355437665783,
0.899580017683466, 0.8933244916003537]
[cometnet-10-21-30-141:6375 hw2 liurenchi$ python logisticR.py
automatically add path:
train/ham
train/spam
test/ham
test/spam
[0.7712201591511936, 0.8327586206896551, 0.7712201591511936, 0.8327586206896551
0.786604774535809, 0.8442970822281167]
[0.7789124668435013, 0.8404509283819629, 0.7789124668435013, 0.8404509283819629
0.8317860300618921, 0.863527851458886]
[0.794761273209549, 0.8519893899204245, 0.8519893899204245, 0.8827586206896552,
0.9279398762157383, 0.9164014146772768]
[cometnet-10-21-30-141:6375 hw2 liurenchi$ python logisticR.py
automatically add path:
train/ham
train/spam
test/ham
test/spam
[0.808709106984969, 0.8327586206896551, 0.808709106984969, 0.8366047745358091,
.8510167992926614, 0.8596816976127322]
[0.808709106984969, 0.8366047745358091, 0.8240937223695844, 0.8519893899204245,
0.9058355437665783, 0.9048629531388153]
[0.8317860300618921, 0.863527851458886, 0.9149646330680814, 0.9072723253757736,
0.5461538461538462, 0.9164014146772768]
cometnet-10-21-30-141:6375 hw2 liurenchi$
```