Logistic Regression

Logistic regression is a discriminative classifier which uses gradient ascent algorithm in order to learn weights that maximize the probability of P(y|x). There is no closed form solution available for logistic regression, hence we concave function and optimize it using gradient ascent. We use MCAP regularization which uses quadratic penalty in order to avoid very large weights. This helps avoid overfitting.

Analysis

1. When we add stopWords file and exclude stop Words while doing the training as well as prediction, we get better accuracy for test set which definitely means that its a good idea to remove stop words before training the model
2. For 300 iterations and lambda=0.01, when we set learning Rate = 0.9, it misses the global maxima and with new weights, we get lesser accuracy on test set (88.7029288702929 instead of 89.12), This means that it may have missed the maxima
3. When we set lambda = 0.01, learning Rate/step = 0.01 and set number of iterations as 1000, we get Test accuracy of 90.1673640167364, which is better than 89.12. This means more number of iterations lead to faster convergence towards the global maxima.
4. When we set number of iterations as 10, then the test set accuracy comes out to be 62.76150627615063, which is very small, this means that for algorithm to converge, we need to execute substantial number of iterations
5. We also see that when we add stopwords, prediction accuracy on training data set reduces to 99% from 100%, which means that stop words also help us reduce overfitting.

Please find statistics for different values of lambda detailed below.

- Learning Rate - 0.05
- Number of iterations - 200

Lambda	Accuracy on Test Set	Accuracy on Test set after stop words removal
0.0001	89.33054393305439	91.42259414225941
0.001	89.1213389121339	90.7949790794979
0.00001	89.33054393305439	92.25941422594143

0.1	90.1673640167364	91.2133891213389
0.5	89.1213389121339	91.42259414225941
1	90.1673640167364	90.7949790794979

Statistics with 300 iterations, lambda = 0.01, learningRate/step = 0.01

```
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\train\ham
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\train\spam

Total accuracy found : 99.35205183585313
Prediction on Test data:
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\test\ham
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\test\spam

Total accuracy found : 89.1213389121339

Stopwords File found: Loaded 499 stop words in memory ( Stats using stop words file)
Training on
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\train\ham
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\train\spam

Prediction on Training data:
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\train\ham
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\train\spam

Total accuracy found : 99.56803455723542
Prediction on Test data:

S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\test\ham
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification\target\classes\test\spam

Total accuracy found : 92.25941422594143
```

How to execute Logistic Regression:

Execute the logisticRegression.Jar, Provide path to training folder, test set folder and stopWords file and other inputs like number of iterations, lambda and learning RAte. It executes the Logistic Regression classifier and prints the accuracy found on the set. It also executes the trained model on training set again and confirms the accuracy on training set in order help us get insight into whether it is overfitting the data or not.

Please find detailed steps for executing the logistic regression jar below.

Steps:

1. Extract the zip file
2. Goto the path where logisticRegression.jar is located in the folder
3. Execute the command “java -jar logisticRegression.jar“
4. Provide it path to the “train” folder which has ham/spam folders for classification.
5. Provide number of iterations
6. Provide lambda
7. Provide learning rate
8. Provide path to test folder with same folder structure as that of train
9. Provide path to stopWords file
10. As you can see in following execution log, it prints accuracy on all data sets it comes across.

Execution Log

```
S:\ML\Source_code\Machine_learning\LogisticRegressionClassification>java -jar logisticRegression.jar
Please provide path to the training folder(should contain two subfolders ham & spam with input files):
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train
number of Iterations:
100
lambda:
0.1
learningRate/Step:
0.1
Training on

S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\ham
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\spam
Prediction on Training data:
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\ham
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\spam
Total accuracy found : 96.11231101511879
Please provide path to Test folder on which prediction needs to be made:
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\test
Prediction on Test data:
```

```
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\test\ham
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\test\spam
Total accuracy found : 86.19246861924687
Please provide path to stopWords file:
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\stopwords.txt
Stopwords File found: Loaded 499 stop words in memory
Training on
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\ham
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\spam
Prediction on Training data:
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\ham
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\train\spam
Total accuracy found : 97.8401727861771
Prediction on Test data:
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\test\ham
S:\ml\Source_code\Machine_learning\LogisticRegressionClassification\src\main\resources\test\spam
Total accuracy found : 90.5857740585774
```