Assignment 2: Text Classification

1 Description of the Assignment

Stack Exchange is a very popular Q&A (Question-and-Answer) based website. We want to analyze some archived data of Stack Exchange (https://archive.org/details/stackexchange) using text classification.

The goal of text classification is to identify the topic for a piece of text (news article, web-blog, etc.). Text classification has obvious utility in the age of information overload, and it has become a popular turf for applying machine learning algorithms. In this project, you will have the opportunity to implement k-nearest neighbor and Naive Bayes, apply these to text classification on Stack Exchange sample data, and compare the performances of these techniques.

1.1 k-Nearest Neighbor (k-NN)

1. Implement the k-NN algorithm for text classification. Your goal is to predict the topic for N (initially take small number of rows, e.g., 50 rows from each file and later increase the number during final submission) number of texts/rows/documents from each file in the Test folder. Try the following distance or similarity measures with their corresponding representations:

   - Hamming distance: each document is represented as a boolean vector, where each bit represents whether the corresponding word appears in the document.
   - Euclidean distance: each document is represented as a numeric vector, where each number represents how many times the corresponding word appears in the document (it could be zero).
   - Cosine similarity with TF-IDF weights (a popular metric in information retrieval): each document is represented by a numeric vector as in the case of euclidean distance. However, now each number is the TF-IDF weight for the corresponding word (as defined below). The similarity between two documents is the dot product of their corresponding vectors, divided by the product of their norms.

2. Let w be a word, d be a document, and N(d,w) be the number of occurrences of w in d (i.e., the number in the vector as in the case of euclidean distance). TF stands for term frequency, and TF(d,w)=N(d,w)/W(d), where W(d) is the total number of words in d. IDF stands for inverted document frequency, and IDF(d,w)=log(D/C(w)), where D is the total number of documents, and C(w) is the total number of documents that contains the word w; the base for the logarithm is irrelevant, you can use e or 2. The TF-IDF weight for w in d is TF(d,w)*IDF(d,w); this is the number you should put in the vector in Cosine similarity. TF-IDF is a clever heuristic to take into account of the "information content" that each word conveys, so that frequent words like "the" is discounted and document-specific ones are amplified. You can find more details about it online or in standard IR text.

3. You should try k = 1, k = 3 and k = 5 with each of the representations above. Notice that with a distance measure, the k-nearest neighborhoods are the ones with the smallest distance from the test point, whereas with a similarity measure, they are the ones with the highest similarity scores.

4. Output the result of running all the three values of k using all the three k-NN techniques into a single file.

1.2 Naive Bayes (NB)

Implement the Naive Bayes algorithm for text classification. Your goal is to predict the topic for N (initially take small number of rows, e.g., 50 rows from each file and later increase the number during final submission) number of texts/rows/documents from each file in the Test folder. Naive Bayes used to be the de facto method for text classification.

1. Consider all the words of a test text/document/question as independent, then calculate the probability of the statement of being a topic and then pick up the topic which has the highest probability score.
2. Try different smoothing factors (at least 50 different values).
3. Output the result of running all the values of smoothing parameter into a single file.

1.3 Comparison and Report Writing

In this part, you will compare between the performance of k-NN classifier and Nave Bayes classifier for text classification. Follow the steps below:

1. Select the best value of k and best measure of distance/similarity (among three) that gave the best performance (For this selection, consider the number of topic as 4, and take 300-500 rows from each of the training files and 50 rows from each of the test files).
2. Select the best value of α for Naive Bayes (For this selection, consider the number of topic as 4, and take 300-500 rows from each of the training files and 50 rows from each of the test files).
3. Take 4 training files each having 5000 rows and 4 test files each having 50 rows. Now run Naive Bayes (with best α) and k-NN (with best k and best measure) for R number of times.

   - Take 100 rows from each training file, train on these 4*100=400 rows/documents and then test 50 rows of each test file. Compute accuracy for Naive Bayes (with best α) and k-NN (with best k and best measure).
   - In the next run, take next 100 rows from each of the training files and then test 50 rows (this 50 rows are fixed for all the runs) of the test files. Compute the accuracy for both of the algorithms. Do the same process for next runs.

   The preferable value of R is 50. But if your program takes a lot of time for a single run, then consider any value between 10 and 50 for R.
4. Compute mean and standard deviation of the accuracy of R number of runs. Then, compute t-statistic at significance levels of 0.005, 0.01 and 0.05, and compare which algorithm (k-NN or Bayesian) is better. Write the results in a report and submit it.

2 Dataset Description

1. The size of the sample dataset is more 100MB. So, collect it during the sessional class. You can also download the zipped file containing data.
   (link: https://www.dropbox.com/s/1jdct708qk8p6za/Data.zip?dl=0)
2. In the training and test folder, there are respective xml files.
3. The topics.txt contains the name of the topics. For each topic, there should be a training xml file and test xml file in the respective folders.
4. For both training and test type of files, take every line which starts with ”<row” and keep only the ”Body” portion of this row. Consider only this portion as a document (or text) and the name of the file as the topic name.

3 Instructions to Group Work

1. Every group has two members. The 50% completion has to be done individually. But the final submission will be combined. The overall work distribution is as follows:
    - The first member will do the full task of implementing K-NN (all three algorithms).
    - The second member will do the task of implementing NB and final merging of NB with k-NN.
2. Report writing and other tasks have to be done in combine.

4 50% Progress

1. 50% will be shown individually.
2. The task of the first member will be as follows:
    - Parsing all the training and test files.
    - Implementation of Hamming and Euclidean measures for k=1,3 and 5.
3. The task of the second member will be as follows:
    - Parsing all the training and test files.
    - Implementation of Naive Bayes for a fixed smoothing factor.

5 Instructions for Report Writing

1. Write the value of t-statistic at significance levels of 0.005, 0.01 and 0.05, and compare which algorithm (k-NN or Bayesian) is better.
2. Answer the following questions in your own language. Remember that the answer should be according to the program you have written.
    - Among the three different measures of the K-NN, which one shows maximum accuracy? Why does it work better than other two?
    - Which one between the k-NN (the best measure among three) and the NB works better? Why?
3. Never copy the report. Just answer the questions precisely. Make it as simple as possible. Too much description is not needed!

6 Special Instructions

1. Don't Copy anything! If you do copy from internet or from any other person or from any other source, you will be severely punished and it is obvious. More than that, we expect Fairness and honesty from you. Don't disappoint us!
2. The report should be in .docx/.pdf (No hardcopy is required). Write precisely in your own language and keep it as simple as possible.
3. Upload the 50% progress submission in moodle within 6 P.M. of 23th Semptember, 2016 (Friday).
4. Upload the final submission in moodle within 6 P.M. of 30th Semptember, 2016 (Friday).
5. For python and matlab, you may not get supporting softwares in the lab. If you do program in these languages, bring your computer in the sessional.

6. You are allowed to show the assignment in your own laptop during the final submission. But in that case, ensure an internet connection as you have to instantly download your code from the moodle and show it.

7 Instructions for moodle upload

1. Upload the assignment within the specified time. Otherwise, we can't accept it :(
2. For 50% progress, use the following rules:
   - If you write code in a single file, then rename it as <Student_id>_<TASK_NAME>. <extension>. For example, if your student id is 1105123, your task was KNN and you have done in java, then your file name should be ”1105123_KNN.java” (similarly ”1105123_NB.java”) . Make a folder having the same name without extension and put the file under the folder.
   - If you write code in multiple files, then put all the necessary files in a folder and rename it as <Student_id>_<TASK_NAME>. For example, if your student id is 1105123, your task was KNN and you have done in java, then your folder name should be ”1105123_KNN”.
   - Finally, zip it and upload it. Done :)
3. For final submission, use the following rules:
   - If you write code in a single file, then rename it as <Student_id1>_<Student_id2>_<code>. <extension>. For example, if your student ids are 1105123 and 11050124, and you have done in java, then your file name should be ”1105123_1105124_code.java”.
   - If you write code in multiple files, then put all the necessary files in a folder and rename it as <Student_id1>_<Student_id2>_<code>. For example, if your student ids are 1105123 and 1105124, and you have done in java, then your folder name should be ”1105123_1105124_code”.
   - The report name should be <Student_id1>_<Student_id2>_<report>.<extension>. For example, if your student ids are 1105123 and 1105124, and it is in pdf format, then the report name should be ”1105123_1105124_report.pdf”.
   - Finally make a main folder, put the code (whether file or folder) and report in it, and rename the main folder as your <Student_id1>_<Student_id2>_<Programming_ language>. For example, ”1105123_1105124_Java”. Then zip and upload it anyone between you two. Done :)

8 About Version

A version is included with the assignment pdf name. This is because if any information is changed, then the version will be upgraded and the changes will be summarized in this section.

8.1 Changes included in version 2

- The statement ”Try various values for Laplacian smoothing parameter” has been replaced with ”Try different smoothing factors” in the second point of the subsection ”Naive Bayes”.
- A dropbox link for data has been added in first point of the section ”Dataset Description”

8.2 Changes included in version 3

- Comparison and Report Writing (Subsection 1.3) has been changed fully.
- The first point of the Report Writing (Section 5) has been added.