10-405 Assignment 2a:
Naive Bayes using GuineaPig

Due: Monday, Feb. 12 2018 23:59 EST via Autolab

February 14, 2018

Policy on Collaboration among Students

These policies are the same as were used in Dr. Rosenfeld's previous version of 10601 from 2013. The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone, and the student should be ready to reproduce their solution upon request. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved, on the first page of their assignment. Specifically, each assignment solution must start by answering the following questions in the report:

- Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered 'yes', give full details: ________________ (e.g. "Jane explained to me what is asked in Question 3.4")
- Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered 'yes', give full details: ________________ (e.g. "I pointed Joe to section 2.3 to help him with Question 2".

Collaboration without full disclosure will be handled severely, in compliance with CMU's Policy on Cheating and Plagiarism. As a related point, some of the homework assignments used in this class may have been used in prior versions of this class, or in classes at other institutions. Avoiding the use of heavily tested assignments will detract from the main purpose of these assignments, which is to reinforce the material and stimulate thinking. Because some of these assignments may have been used before, solutions to them may be (or may have been) available online, or from other people. It is explicitly forbidden to use any such sources, or to consult people who have solved these problems before. You must solve the homework assignments completely on your own. I will mostly rely on your wisdom and honor to follow this rule, but if a violation is detected it will be dealt with harshly. Collaboration with other students who are currently taking the class is allowed, but only under the conditions stated below.

1 Important Note

In this assignment, **you will be using Python**.

In this assignment, you will have to create a Naive Bayes classifier that will run with GuineaPig on Python. Assignment 2a WILL NOT be graded but you do need to submit your code (need not be fully working) on Autolab by 2/12/2018. The submitted code will be checked manually. This will help you to start early and budget your time. 2b WILL be graded. You should plan to finish 2a before 2b is released so you have time to finish 2b.

Vidhan Agarwal (vidhana@andrew.cmu.edu) and Vivek Shankar (vshanka1@andrew.cmu.edu) are the contact TAs for this assignment. Please post clarification questions to Piazza, and the instructors can be reached at: *10405-Instructors@cs.cmu.edu*.

2 Introduction

In this part of the assignment, you need to implement the Naive Bayes algorithm in GuineaPig. In assignment 2a, the goal is to re-implement the Naive Bayes training algorithm in Guinea Pig. Later, in assignment 2b, you will be implementing a small-memory test algorithm for Naive Bayes - i.e, testing where the vocabulary and test data do not fit into memory. **Note: some counts such as number of labels and vocabulary size cannot be pre-computed and used directly as parameters**.

Below is a high-level sketch of the algorithm. This algorithm includes steps for both training and testing, but note that only the first step pertains to training-assignment 2a. Feel free to get started with the other steps if you finish 2a early!:

- Generate event counts from the training data
- Convert the event counts to key-value pairs (k,v) where k is a word, and v is some representation of all the counts for that word.
- Generate requests for word counts from the test file, i.e. flatten the test documents into pairs(word,docId).
- Join the flattened test documents with the reorganized event counts.
- Group the joined result together so that you can classify the test documents.

2.1 Introduction to GuineaPig

GuineaPig is a lightweight Python library that is similar to Pig, but is easier to learn and debug. You are required to have some working knowledge of Python and Hadoop to be able to use GuineaPig. You will express your workflow using high level constructs (such as Join, Augment etc.) and GuineaPig spawns off MapReduce tasks behind the scenes to do the compute. GuineaPig provides for you a layer of abstraction over bare-bones Hadoop. You can find more information about GuineaPig (including a tutorial) at `http://curtis.ml.cmu.edu/w/courses/index.php/Guinea_Pig`. It is recommended that you read this document before proceeding with the assignment. You can directly download GuineaPig at `https://github.com/TeamCohen/GuineaPig/archive/master.zip` or work with source code distributed with the tutorial. `http://curtis.ml.cmu.edu/w/courses/index.php/Guinea_Pig#Quick_Start`.

2.2 Local Execution

In addition to providing an abstraction over Hadoop, GuineaPig programs can also be executed locally, without Hadoop. This feature makes your program much easier to debug. We recommend that you debug locally and test on Hadoop using the Stoat cluster before submitting.

2.3 Using Hadoop on the Stoat cluster

By now, you should all have access to Stoat, which already has an installation of Hadoop ready to use. You just need to copy GuineaPig, with your code, to the cluster:

```
scp -r /path/to/GuineaPig <username>@shell.stoat.pdl.local.cmu.edu:~
```

You can SSH to Stoat from within CMU?s network:

```
ssh <username>@shell.stoat.pdl.local.cmu.edu
```

In order to set up GuineaPig to execute on Hadoop, follow the instructions on the wiki: `http://curtis.ml.cmu.edu/w/courses/index.php/Guinea_Pig#Using_Hadoop`.

3 The Data

We are using the same dataset as the one in Homework 1. These are articles from DBPedia, and the label is the type of the article. There are in total 18 classes in the dataset, and they are from the first level class in DBpedia ontology. For more information about this dataset, you can refer to `http://wiki.dbpedia.org/Downloads2015-04`. The data is of the format:

```
<id>    <label1>,<label2>,<label3>...    w1 w2 w3 w4...
```

The three columns are separated by tab, and the documents are preprocessed so that there are no tabs in the body. You can find the data in `/afs/cs.cmu.edu/project/bigML/dbpedia_16fall/`.

For this homework, feel free to play with the parameters, tokenizer and output to improve your Naive Bayes classifier.

4 Autolab Implementation Details

We have given you a starter file, nb.py, that contains some setup code. You should implement Naive Bayes in this file, and it should handle Naive Bayes training. We will test your code with the following command (note that you can ignore the test file parameter for 2a):

```
python nb.py --store output \\
    --params trainFile:path/to/train,testFile:path/to/test
```

After running, you should store your final predictions in the "output" variable, which will then be stored in "gpig_views/output.gp" by GuineaPig. The output file should be of the form

```
(event, count)
(event, count)
...
```

where `event` is some representation of the name of a particular Naive Bayes event counter, and count is its corresponding count. You don't need to worry about the ordering of the output, as long as all the information is there.

For the autograding, you don't need to worry about setting Python paths. Just assume that `from guineapig import *` will work.

You should look through the GuineaPig guide at `http://curtis.ml.cmu.edu/w/courses/index.php/Guinea_Pig`. The wordcount example might be a good place to start.

5 Submission

Submit a tarball containing the code and a pdf containing answers to questions mentioned in the policy collaboration section on autolab.

```
tar -cvf hw2a.tar *.py *.pdf
```