10605 BigML Assignment 4(a): Naive Bayes using Hadoop Streaming

Due: Friday, Feb. 21, 2014 23:59 EST via Autolab

Late submission with 50% credit: Sunday, Feb. 23, 2014 23:59 EST via Autolab

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

As usual, **you are expected to use Java for this assignment**.

This assignment is worth 50 points. This is a relatively small assignment because you will be able to reuse most of the code from previous Naive Bayes assignment. In this assignment, you will have to use Hadoop Streaming to train Naive Bayes classifier. To learn more about Hadoop Streaming, see `http://hadoop.apache.org/docs/stable1/streaming.html`.

Siddharth Varia (varias@cs.cmu.edu) is the contact TA for this assignment. Please post clarification questions to the Piazza, and the instructors can be reached at the following email address: *10605-Instructors@cs.cmu.edu*.

2 Introduction

In Hadoop Streaming, the input is read from stdin, and output is written to stdout. The mapper takes input coming from stdin and outputs key/value pairs. Hadoop's Shuffle and Sort phase takes care of sorting keys so that input feed to reducers will be sorted by keys. The reducer aggregates values corresponding to the same key and outputs key/value pairs to stdout. The mapper and reducer need not be java classes, they can even be an executable or a script.

Specifically, in the homework 4(a), a successful Hadoop Streaming naive Bayes implementation will need to have a mapper class that counts the occurrences of terms and output the key value pairs, as well as a reducer that merges these counts of the same keys. Note that you only need to run Hadoop streaming to train your naive Bayes classifier, and no testing is involved in this part of homework 4.

3 Getting started

3.1 Using AWS and elastic MapReduce (EMR)

1. Sign up for AWS using your credit card[1]

[1] You may check your usage at `https://console.aws.amazon.com/billing/home#/bill`. Note that job less than an hour will be billed in an full hour.

2. Redeem AWS code[2] provided to you (the gift code will be emailed to you in a separate email).

3. Create new AWSAccessKeyId and AWSSecretKey by going to Security Credentials and save key file.

4. Refer to AWS EMR documentation on how to use it through commandline or web console. see `http://aws.amazon.com/elasticmapreduce/getting-started/` and next subsection for more details.

We will distribute the AWS gift code to every registered student. If you have not got one, let us know. Here are a few hints for running jobs on AWS:

3.1.1 Submitting a Jar job

The easiest way to submit jobs to AWS is via the AWS console's web interface.
For details, see `http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-what-is-emr.html`.
You can also start a job using command line interface. You need to install the elastic mr command line interface.
For details, see `http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-cli-reference.html`

Important: in homework 4a, when setting up your job on EMR, make sure you select "**Streaming program**" in the add step option, so that the EMR will run your job in the Hadoop streaming mode.

3.1.2 Viewing job progress

Tutorial for viewing the jobtracker on your local machine (via proxy) [3]. (You can also ssh into the machine using the command line interface, and then use the lynx commands in the login preamble to view the job tracker.)

3.2 Debugging with the CMU Hadoop cluster

To help you debugging your Hadoop streaming code, you may debug on the hadoop cluster at CMU. See the course webpage for details: `http://curtis.ml.cmu.edu/w/courses/index.php/Hadoop_cluster_information`. Another option would be setting up the Hadoop environment on your local machine and simulate the large jobs by running a single thread with a small file locally.

[2] Please do not use other's gift code or let others use your gift code. If you have problems, let the instructors know. Also, please do not sell the gift code on Amazon or eBay.

[3] `http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/UsingtheHadoopUserInterface.html`

4 The Data

We are using the data from Assignments 1. They are on AWS at `s3://bigmldatasets/rcv1/full/` and `s3://bigmldatasets/rcv1/small/`. You do not need to copy the data to S3 by yourself.

Note that you only need to submit the log files for the **full dataset**. The small dataset is provided to you for debugging as usual[4].

Similar to homework 1, please use the provided tokenizer from homework 1, and use only the four valid labels specified in homework 1. Also similar to homework 1, if the document contains multiple valid labels, then you have to consider the document once for each such label.

5 Deliverables

5.1 Steps

What you need to do in this assignment can be summarized in the following steps:

- Setup you AWS account following the instructions in the Section 3.1 of the handout.
- Upload your jar file that contains your mapper and reducer for naive Bayes training to S3.
- Setup and run the job on **full dataset** with elastic MapReduce using the Streaming option.
- Download the controller and syslog text files, and submit via Autolab together with the report and your code.

5.2 Report

Submit your implementations via AutoLab. You should implement the algorithm by yourself instead of using any existing machine learning toolkit. You should upload your code (including all your function files) along with a **report**, which should solve the following questions:

1. Examine the output of your Hadoop streaming job on the full RCV1 training dataset. What are the counts for the following keys? (20 points)

   ```
   Y=GCAT,W=gunman ?
   Y=GCAT,W=half ?
   Y=CCAT,W=limited ?
   ```

[4]Note that Amazon will charge you a full hour for small jobs less than one hour, so you may not want to develop or debug your code on AWS.

2. Answer the questions in the collaboration policy on page 1.

The controller and syslog files **from AWS** for the mapreduce job can be downloaded from the AWS console. Simply go to the Elastic Mapreduce tab. Select your job in the list, click View details and expand Steps to see jobs and log files.

5.3 Autolab submission details

You should tar the following items into **hw4a.tar** and submit to the homework 4a assignment via Autolab:

- NBTrainMapper.java
- NBTrainReducer.java
- controller.txt
- syslog.txt
- and all other auxiliary functions you have written
- report.pdf

Tar the files directly using "tar -cvf hw4a.tar *.java *.txt report.pdf". Do **NOT** put the above files in a folder and then tar the folder. You do not need to upload the saved temporary files.

6 Submission

You must submit your homework through Autolab. In this part of homework 4, there will be no validation needed.

- Homework4a: This is where you should submit your tar ball. You have a total of **5 possible submissions**. Your score will be reported, and feedback will be provided immediately.

7 Grading

If you are able to successfully run the job on AWS EMR, you will receive 30 points. The successful run of your job should be reflected in your log files. The report will be graded manually (20 points).