10605 BigML Assignment 4(b): Naive Bayes with Hadoop API

Due: Friday, Feb. 28, 2014 23:59 EST via Autolab

Late submission with 50% credit: Sunday, Mar. 2, 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. Similar to part (a), this part (b) is also a relatively small assignment because you will be able to reuse some of the code from previous Naive Bayes assignments. However, unlike the Hadoop streaming settings in part (a), in this assignment, you will have to port your naive Bayes code to the real Hadoop environment using Hadoop APIs to train a naive Bayes classifier.

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 this part of the assignment, you need to re-implement naive Bayes for the Hadoop MapReduce framework. Similar to part (a), you only need to write Hadoop naive Bayes training, and you do not need to care about the testing part.

2.1 Using AWS and elastic MapReduce (EMR)

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

2.1.1 Submitting a Jar job

Important: unlike homework 4a, in this part b, when setting up your job on EMR, make sure you select "**Custom Jar**" in the add step option, so that the EMR will run your job in the Hadoop API mode.

2.1.2 Viewing job progress

Tutorial for viewing the jobtracker on your local machine (via proxy) [1]. (You can also ssh into the machine using the command line interface, and then use the lynx commands in

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

the login preamble to view the job tracker.)

2.2 Debugging with the CMU Hadoop cluster

To help you debugging your Hadoop 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.

2.3 Additional Hadoop Tutorial

In case you want to study extra tutorials about Hadoop, your honorary TA Malcolm Greaves has kindly put together a wiki page here: `http://curtis.ml.cmu.edu/w/courses/index.php/Guide_for_Happy_Hadoop_Hacking`

3 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 running your code on the **full dataset**. The small dataset is provided to you for debugging as usual[2].

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.

4 Deliverables

4.1 Steps

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

- Port the naive Bayes training code into Hadoop using Hadoop's MapReduce API.
- Run the Hadoop API MapReduce job on AWS with the **full dataset** with elastic MapReduce using the Custom Jar option.
- Download the controller and syslog text files, and submit via Autolab together with the report and your source code in a tar ball.

[2]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.

4.2 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. Compare the Hadoop Streaming mode with the Hadoop API. What are the major differences? (5 points)
2. For the streaming and non-streaming modes, when would you choose one over the other? (5 points)
3. In the Hadoop version of naive Bayes, how would you estimate the vocabulary size? (5 points)
4. How would you design the testing pipeline of large-scale naive Bayes classification using Hadoop? (5 points)
5. 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.

4.3 Autolab Implementation details

Like the Gates clusters, Autolab is currently running Hadoop 1.0.1. In this part of homework, you will need to follow the exact naming of the following files.

You must have the **run.java** class, which includes the main function that you call the MapReduce version of the naive Bayes trainer (e.g. NB_train_hadoop.java). There will be three arguments sent to this main function: **InputPath**, **OutputPath**, and **the number of reduce tasks**. For example, in a standalone debugging version of Hadoop, your code need to execute sucessfully via the following commands:

```
javac -cp hadoop-core-1.0.1.jar:. *.java;
java -cp hadoop-core-1.0.1.jar:hadoop/lib/*:. run InputPath OutputPath 1
```

Important: you should still use the tokenizer provided in homework 1, and also the key output format mentioned in homework 1. For example:

```
Y=CCAT,W=he 3.0
Y=CCAT,W=saw 1.0
Y=CCAT,W=her 3.0
Y=CCAT,W=duck 4.0
```

```
Y=CCAT,W=or 1.0
Y=CCAT,W=* 123.0
Y=CCAT 10.0
Y=* 10.0
...
```

But this time Hadoop will do the sorting job for you.

You should tar the following items into **hw4b.tar** and submit to the homework 4b assignment via Autolab:

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

Tar the files directly using "tar -cvf hw4b.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.

5 Submission

You must submit your homework through Autolab. In this part of homework 4, there will be a validation link.

- Homework4b-validation: You will be notified by Autolab if you can successfully finish your job on the Autolab virtual machines. Note that this is not the place you should debug or develop your **algorithm**. All development should be done on linux.andrew.cmu.edu machines. This is basically a Autolab debug mode. There will be **NO** feedback on your **performance** in this mode. You have unlimited amount of submissions here. To avoid Autolab queues on the submission day, the validation link will be closed 24 hours prior to the official deadline. If you have received a score of 1000 with no errors, this means that you code has passed the validation.
- Homework4b: 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.

6 Grading

If you are able to successfully run the job on full dataset with AWS EMR, you will receive 10 points. The successful run of your job should be reflected in your log files. We will test your Hadoop code on Autolab in real time, and check the log and the correctness of your output (20 points). The report will be graded manually (20 points).