Text Classification using Mahout

(Nov. 6, 2012)

1. Module name: Text Classification using Mahout

2. Scope: This module focuses on classification of text using Apache Mahout

3. Learning objectives

After successful completion of this module, students will be able to do the following:

a. explain and apply methods of text classification

b. correctly classify a set of documents using Apache Mahout

c. construct and apply workflows for text classification using Apache Mahout

4. 5S characteristics of the module

a) **Streams:** Streams include data and different types of results, such as text file providing classification information, etc.

b) **Structures:** Sample document collections distributed with Apache Mahout are in CSV format. Other formats can be used after preprocessing.

c) **Spaces:** The indexed documents and queries are logically represented as vectors in a vector space. The document collections are physically stored on the server running LucidWorks.

d) **Scenarios:** Scenarios include users classifying a collection of documents into different classes. As an example, a collection of emails can be classified as spam or non-spam.

e) **Society:** This module can be used by those who intend to classify a collection of documents. Potential end users are researchers, librarians, or any people who wish to learn Apache Mahout for text classification.

5. **Level of effort required**

The required amount of time to complete this module should be about 4-5 hours.

a. **Out-of-class:** 4-5 hours

b. **In-class:** Students may ask questions and discuss exercises with their teammates.

6. **Relationships with other modules (flow between modules)**

The module is not directly related to any other previous modules. The classification results from Apache Mahout can be used in different existing tools (e.g., NLTK) for information retrieval and analysis.

7. **Prerequisite knowledge/skills required (what the students need to know prior to beginning the module; completion optional; complete only if prerequisite knowledge/skills are not included in other modules)**

a. Textbook Ch. 13 (Text Classification)

b. Mahout in Action, Chs. 13-17

c. Basic UNIX commands.

8**. Introductory remedial instruction (the body of knowledge to be taught for the prerequisite knowledge/skills required; completion optional)**

Students should know about at least the following: Text Classification, also called text categorization, is a problem in information retrieval. The general goal of text classification is to assign a given object to topic(s) which it belongs to, based on previously created rules, for example, to assign an email into 'spam' or 'non-spam' or to assign a book into 'fiction' or 'history'. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. Classification normally refers to a supervised procedure, e.g., a procedure that learns to classify new instances based on learning from a training set of instances that have been properly labeled manually with the correct classes.

9**. Body of knowledge:**

A. Naïve Bayes Text Classifier

a. Classification rule:

$$c_{map} = \arg\max_{c \in C}[\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)]$$

b. Simple interpretation:

i. Each conditional parameter $\log \hat{P}(t_k|c)$ is a weight that indicates how good an indicator term, t_k , is for class, c.

ii. The prior $\log \hat{P}(c)$ is a weight that indicates the relative frequency of class c.

iii. The sum of log prior and term weights is then a measure of how much evidence there is for the document being in the class.

iv. We select the class with the most evidence.

B. Introduction to Mahout

Apache Mahout is a scalable machine learning library. Mahout can be used on a wide range of classification projects, but the advantage of Mahout over other approaches becomes striking as the number of training examples gets extremely large. What large means can vary enormously. Up to about 100,000 examples, other classification systems can be efficient and accurate. But generally, as the input exceeds 1 to 10 million training examples, something scalable like Mahout is needed.

I. Mahout Login Details

You can login with the following credentials and information via ssh from a terminal. Mahout 0.7 is already installed on the host machine.

```
hostname: xxx.xxx.xxx.xxx
username: ********
password: **************
```

The installation directory for mahout: *$HOME/mahout-distribution-0.7*

Mahout executable is located at: *$HOME/mahout-distribution-0.7/bin*

Sample data set (20-Newsgroups data) provided at: *$HOME/mahout-distribution-0.7/20news-bydate*

II. HADOOP

HADOOP version 0.20.205.0 is already installed on the host machine at: *$HOME/hadoop-0.20.205.0*
You need to add the following command at the beginning of the shell script:
HADOOP_HOME=/home/CS5604/hadoop-0.20.205.0/bin

C. Typical Workflow of Text Classification

In this section, we will provide step-by-step instructions (commands) to classify a set of documents using Mahout. For the purpose, we are using the 20-Newsgroup data, and Naïve Bayes is used as classifier. The examples below describe what we have done. **Do not do this too, since there is a shared space, and executing these commands will overwrite the work of other groups. See #10 below for instructions on exercises you should do, putting working results in directories for your team.**

a. Create a working directory and copy data

Command:

```
mkdir 20news-all

cp –R 20news-bydate/*/* 20news-all
```

Here, we are creating a directory named 20news-all and copying all the data into this directory.

b. Creating sequence files from 20newsgroups data

Command:

```
./bin/mahout seqdirectory -i 20news-all -o 20news-seq
```

As an intermediate data format, we need to convert input data to a sequence file. In this command, '-i' denotes the specified input file/directory while '-o' denotes the output file/directory.

c. **Converting sequence files to vectors**

Command:

```
./bin/mahout seq2sparse \
    -i 20news-seq \
    -o 20news-vectors  -lnorm -nv  -wt tfidf
```

Here,

`--logNormalize` `-lnorm`	`(Optional) Whether output vectors should be logNormalize. If set true else false`
`--namedVector` `-nv`	`(Optional) Whether output vectors should be NamedVectors. If set true else false`
`--weight <weight>` `-wt <weight>`	`The kind of weight to use. Currently <weight> = TF or TFIDF`

With the new standard for Mahout, the vector data format is preferred to old formats. The above command converts previously formatted sequence files into vectors.

d. **Creating training and holdout set with a random 80-20 split of the generated vector dataset**

Command:

```
./bin/mahout split \
    -i 20news-vectors/tfidf-vectors \
    --trainingOutput 20news-train-vectors \
    --testOutput 20news-test-vectors  \
    --randomSelectionPct 20 --overwrite --sequenceFiles
-xm sequential
```

Here,

`--overwrite` `-ow`	If present, overwrite the output directory before running job
`--sequenceFiles` `-seq`	Set if the input files are sequence files. Default is **false**

`--method <method>`
`-xm <method>`
	The execution method to use:
	`<method>` = **sequential** or **mapreduce.**
	Default is **mapreduce**

In practice, training examples are typically divided into two parts. One part, known as the training data, consists of 80–90 percent of the available data. The training data is used in training to produce the model. A second part, called the test data, is then given to the model without telling it the desired answers, although they're known. This is done in order to compare the output that results from classifying, using the desired output. Here, we split the data set randomly as 80% training data and 20% test data. The value 20 after the "randomSelectionPct" parameter indicates that 20% of the data will be selected randomly for the test data set.

e. Training Naive Bayes model

Command:

```
./bin/mahout trainnb \
    -i 20news-train-vectors -el \
    -o model \
    -li labelindex \
    -ow
```

Here,

`--labelIndex <labelIndex>`
`-li <labelIndex>`
	The path to store the label index in

`--overwrite` `-ow`	If present, overwrite the output directory before running job

This is the actual command for training the Naïve Bayes Classifier. The command takes as input the previously selected set of training data and outputs a model.

f. Self testing on training set

Command:

```
./bin/mahout testnb \
    -i 20news-train-vectors\
    -m model \
    -l labelindex \
    -ow -o 20news-testing
```

Here,

--model <model> -m <model>	The path to the model built during training
--labelIndex <labelIndex> -l <labelIndex>	The path to the location of the label index
--overwrite -ow	If present, overwrite the output directory before running job
--output <output> -o <output>	The directory pathname for output

This command tests the training data set using the classifier model previously built. We expect almost 100% accuracy here because testing has been done on the same training set that was used for training the model.

g. Testing on holdout set

Command:

```
 ./bin/mahout testnb \
    -i 20news-test-vectors\
    -m model \
    -l labelindex \
    -ow -o 20news-testing $c
```

Based on the learning experience, the classifier (model) is now applied to the actual test data. The output of this command also shows the accuracy of classification.

10. Exercises / Learning activities

[a] For this exercise, students need to classify test data using Mahout.

i. Write down a single shell script file that will execute all the above commands (a-g) to classify the 20-Newsgroup sample data for the following 3 scenarios using Naïve Bayes classifier. The initial working directory name will be the same as *team's project name –sub bullet number*. For example, the folder names will be "ProjCINETViz-1", "ProjCINETViz-2" and "ProjCINETViz-3" for our group.

   1) For this exercise, use the sample data under the directory "mahout/20news-bydate". Split the data set randomly as 80% training data and 20% test data (holdout set) and summarize the results by giving the number of total classified instances and the percentage of accuracy.

   2) For this exercise, use the sample data under the directory "mahout/20news-bydate/20news-bydate-train". Split it randomly as 80% training data and 20% test data for a second run and summarize the results.

   3) For this exercise, use the sample data under the directory "mahout/20news-bydate/20news-bydate-test". Split it randomly as 80% training data and 20% test data for a third run and summarize the results.

ii. Compare the results as per the above discussion for the three classification tasks and comment on the variation in the accuracy. Is the classifier a good/fair/bad one? Justify your answer.

iii. Is it possible that the classification results' accuracy varies significantly for different sets of test data? Why, or why not?

[b] **Textbook exercise 13.9:** (Please provide the detailed steps to reach the classification)

Based on the data in Table 13.10 of the textbook (copied below),

(i) estimate a multinomial Naive Bayes classifier,
(ii) apply the classifier to the test document,
(iii) estimate a Bernoulli NB classifier,
(iv) apply the classifier to the test document.

You need not estimate parameters that you don't need for classifying the test document.

Table 13.10 of textbook: Data for exercise 10[b] (textbook exercise: 13.9)

	docID	**words in document**	**in c = China?**
training set	1	Taipei Taiwan	Yes
	2	Macao Taiwan Shanghai	Yes
	3	Japan Sapporo	No
	4	Sapporo Osaka Taiwan	No
test set	5	Taiwan Taiwan Sapporo	**?**

11. Evaluation of learning objective achievement

The evaluation of learning objectives shall be done based on the following: (1) the correctness of the procedure and calculation of exercise classification tasks, and (2) the correctness of the workflow while classifying text collections using Mahout.

12. Resources.

[a] Mahout in Action, by Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman, Manning Publications Co., 2011.

[b] LucidWorks documentation, URL: http://lucidworks.lucidimagination.com/display/bigdata/Document+Retrieval

[c] 20news-group dataset source, URL: http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate.tar.gz

[d] Manning, C., Raghavan, P., and Schutze, H. (2008). Chapter 13: Text Classification. In Introduction to Information Retrieval. Cambridge: Cambridge University Press.

13. Glossary

Text Classification; Training data; Test data; Naïve Bayes

14. Additional useful links

a) http://mahout.apache.org/

15. Contributors

Authors: Maksudul Alam (maksud@vt.edu), S M Arifuzzaman (sm10@vt.edu) and Md Hasanuzzaman Bhuiyan (mhb@vt.edu)

Reviewers: Dr. Edward A. Fox, Kiran Chitturi, Tarek Kanan
Class: CS 5604: Information Retrieval and Storage. Virginia Polytechnic Institute and State University