E6893 Big Data Analytics:

Demo Session for Classification

Ruichi Yu

Oct 30, 2014
Mahout Classification

Mahout provides:

1. Naive Bayes
2. Hidden Markov Models
3. Logistic Regression
4. Random Forests
Naive Bayes classification:

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.
Naive Bayes classification example:

Twenty Newsgroups Classification Example
Naive Bayes classification example:

The 20 newsgroups dataset is a collection of approximately 20,000 newsgroup documents, evenly across 20 different newsgroups. Mahout CBayes classifier to create a model that would classify a new document into one of the 20 newsgroups.
Prerequisites:

Mahout
Maven
Part 1: Use existed model

1. Download Mahout:
https://mahout.apache.org/general/downloads.html

2. Download Mahout-trunk:
git clone git://git.apache.org/mahout.git mahout-trunk

3. For Maven users please include the following snippet in your
pom under mahout-trunk folder:
<dependency>
   <groupId>org.apache.mahout</groupId>
   <artifactId>mahout-core</artifactId>
   <version>${mahout.version}</version>
</dependency>
Part 1: Use existed model

4. If running Hadoop in cluster mode, start the hadoop daemons by executing the following commands:
   $ cd $HADOOP_HOME/bin
   $ ./start-all.sh

Running locally:
$ export MAHOUT_LOCAL=true

5. Before running, please make sure you have already set up javahome
   export JAVA_HOME=/Library/Java/Home

6. In the trunk directory of Mahout, compile and install Mahout:
   $ cd $MAHOUT_HOME
   $ mvn -DskipTests clean install
Part 1: Use existed model

7. Run the 20 newsgroups example script by executing:

```
$ ./examples/bin/classify-20newsgroups.sh
```

8. Please select the algorithm you would like to use. Here we choose 1.

Then you can see the results.
### Part 1: Use existed model

#### Statistics

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kappa</td>
<td>0.8549</td>
</tr>
<tr>
<td>Accuracy</td>
<td>88.9064%</td>
</tr>
<tr>
<td>Reliability</td>
<td>84.4083%</td>
</tr>
<tr>
<td>Reliability (standard deviation)</td>
<td>0.2217</td>
</tr>
<tr>
<td>Weighted precision</td>
<td>0.8891</td>
</tr>
<tr>
<td>Weighted recall</td>
<td>0.8891</td>
</tr>
<tr>
<td>Weighted F1 score</td>
<td>0.8864</td>
</tr>
</tbody>
</table>
Part 2: Train your own model

1. Set up your path: (very important)

   export MAHOUT_HOME=/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/bin

   export MAHOUT_CONF_DIR=/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/src/conf

2. Build your working directory

   export WORK_DIR=/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/WorkDir

   mkdir -p ${WORK_DIR}
Part 2: Train your own model

3. Download and extract the 20news-bydate.tar.gz from the 20newsgroups dataset to the working directory:

```bash

mkdir -p ${WORK_DIR}/20news-bydate
cd ${WORK_DIR}/20news-bydate && tar xzf ../*
cd .. && cd ..
mkdir ${WORK_DIR}/20news-all
cp -R ${WORK_DIR}/20news-bydate/*/* ${WORK_DIR}/20news-all
```
Part 2: Train your own model

4. Convert the full 20 newsgroups dataset into a <Text, Text>:
SequenceFile is a hadoop class which allows us to write arbitrary (key, value) pairs into it

Important Hint here:
Please use the full path of mahout!!

/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/bin/mahout
seqdirectory -i ${WORK_DIR}/20news-all -o ${WORK_DIR}/20news-seq -ow
Part 2: Train your own model

5. Convert and preprocesses the dataset into a `<Text, VectorWritable>` SequenceFile containing term frequencies for each document:

/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/bin/mahout seq2sparse -i ${WORK_DIR}/20news-seq -o ${WORK_DIR}/20news-vectors -lnorm -nv -wt tfidf
Part 2: Train your own model

6. Split the preprocessed dataset into training and testing sets:

```
/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/bin/mahout split -i ${WORK_DIR}/20news-vectors/tfidf-vectors --trainingOutput ${WORK_DIR}/20news-train-vectors --testOutput ${WORK_DIR}/20news-test-vectors --randomSelectionPct 40 --overwrite --sequenceFiles -xm sequential
```
Part 2: Train your own model

7. Train the classifier:

Important Hint here: abc is the path you store the labelindex. You can change it to other name

/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/bin/mahout trainnb -i ${WORK_DIR}/20news-train-vectors -el -o ${WORK_DIR}/model -li ${WORK_DIR}/abc -ow -c
Part 2: Train your own model

8. Test the classifier:
/Users/Rich/Documents/Courses/Fall2014/BigData/mahout-distribution-0.9/mahout-trunk/bin/mahout testnb -i ${WORK_DIR}/20news-test-vectors -m ${WORK_DIR}/model -l ${WORK_DIR}/abc -ow -o ${WORK_DIR}/20news-testing -c
Part 2: Train your own model

```plaintext
0 0 0 0 0 0 0 0 0 0 0 0 1 1 5 426
0 2 1 0 0 0 0 0 0 0 0 1 | 435
0 2 2 0 1 1 0 0 1 | 406
1 2 0 5 6 2 10 | 388
1 0 1 1 | 394
1 1 0 0 2 2 1 | 381
1 1 1 2 1 0 0 1 | 381
2 4 1 4 0 0 0 0 0 | 381
0 1 0 0 2 2 | 397
1 0 0 0 1 1 2 2 | 390
1 361 2 9 | 390
21 1 0 0 0 0 0 0 0 1 | 239
2 4 5 187 4 | 239
2 0 1 1 0 0 1 2 2 | 335
4 16 3 295 | 335
```

Statistics

- Kappa: 0.8576
- Accuracy: 89.2947%
- Reliability: 84.8391%
- Reliability (standard deviation): 0.216
- Weighted precision: 0.8941
- Weighted recall: 0.8929
- Weighted F1 score: 0.8908