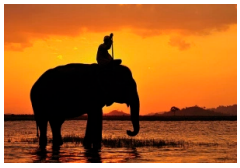
Playing with the Mahout recommendation engine on a Hadoop cluster

2013/02/20 27 COMMENTS (HTTPS://CHIMPLER.WORDPRESS.COM/2013/02/20/PLAYING-WITH-THE-MAHOUT-RECOMMENDATION-ENGINE-ON-A-HADOOP-CLUSTER/#COMMENTS)



Apache Mahout (http://mahout.apache.org/) is an open source library which implements several scalable machine learning algorithms. They can be used among other things to categorize data, group items by cluster, and to implement a recommendation engine.

In this tutorial we will run the Mahout recommendation engine on a data set of movie ratings and show the movie recommendations for each user.

For more details on the recommendation algorithm, you can look at the tutorial from Jee Vang (http://www.slideshare.net/vangjee/a-quick-tutorial-on-mahouts-recommendation-engine-v-04).

Requirement

- Java (to run hadoop)
- Hadoop (used by Mahout)
- Mahout
- Python (use to show the result)

Running Hadoop

In this section, we are going to describe how to quickly install and configure hadoop on a single machine.

Alternatively you can follow the instruction on this post (https://chimpler.wordpress.com/2013/01/20/deploying-hadoop-on-ec2-with-whirr/) to deploy hadoop for free on an Amazon EC2 cluster.

To install Hadoop on your local box, go to http://www.apache.org/dyn/closer.cgi/hadoop/common/ (http://www.apache.org/dyn/closer.cgi/hadoop/common/) and download hadoop-1.1.1.tar.gz
Uncompress the archive:

```
tar xvfz hadoop-1.1.1-bin.tar.gz
```

Edit the file conf/hadoop-env.sh and add the following line:

```
export JAVA_HOME=<JDK DIRECTORY>
```

Generate a rsa key to ssh to your local box without password:

```
ssh-keygen -t rsa -P ''
```

And save in <HOME>/.ssh/id_rsa
Now authorize the access to your local box to itself

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Check that it works by doing

```
ssh localhost ls
```

It should not ask for your password.

If you don't have a ssh server installed, you can install it by typing:

```
sudo apt-get install openssh-server
```

Now set the environment variables:

```
export HADOOP_PREFIX=<HADOOP DIRECTORY>
export HADOOP_CONF_DIR=$HADOOP_PREFIX/conf
export PATH=$HADOOP_PREFIX/bin:$PATH
```

To configure HDFS, edit the file conf/core-site.xml and add the following property in configuration:

```
<configuration>
        <property>
                <name>fs.default.name</name>
                <value>hdfs://localhost:9000</value>
        </property>
</configuration>
```

Then format the HDFS filesystem:

```
hadoop namenode -format
```

We are now ready to start hadoop:

```
start-all.sh
```

Mahout

To install Mahout, go to http://www.apache.org/dyn/closer.cgi/mahout/ (http://www.apache.org/dyn/closer.cgi/mahout/) and download mahout-distribution-0.7.tar.gz

Uncompress the archive:

```
tar xvfz mahout-distribution-0.7.tar.gz
```

Getting the movie dataset

The recommender engine accepts any files containing a set of lines with the userId, the itemId and a preference value(optional) separated by a tab. The userId and itemId must be an integer and the preference value can be an integer or a double.

The GroupLens Movie DataSet (http://grouplens.org/node/73) provides the rating of movies in this format. You can download it: MovieLens 100k (http://www.grouplens.org/system/files/ml-100k.zip).

Uncompress the archive

```
unzip ml-100k.zip
```

This archive contains:

- u.data: contains several tuples(user_id, movie_id, rating, timestamp)
- u.user: contains several tuples(user_id, age, gender, occupation, zip_code)
- u.item: contains several tuples(movie_id, title, release_date, video_release_data, imdb_url, cat_unknown, cat_action, cat_adventure, cat_animation, cat_children, cat_comedy, cat_crime, cat_documentary, cat_drama, cat_fantasy, cat_film_noir, cat_horror, cat_musical, cat_mystery, cat_romance, cat_sci_fi, cat_thriller, cat_war, cat_western)

This data set contains 943 users, 1,682 movies and 100,000 ratings.

Running the Mahout recommender

First we need to copy the file u.data to HDFS:

```
cd ml-100k
hadoop fs -put u.data u.data
```

To run the mahout recommender, type:

```
hadoop jar <MAHOUT DIRECTORY>/mahout-core-0.7-job.jar org.apache
```

With the argument "-s SIMILARITY_COOCURRENCE", we tell the recommender which item similary formula to use. With SIMILARITY COOCURRENCE, two items(movies) are very similar if they often appear together in users' rating. So to find the movies to recommend to a user, we need to find the 10 movies most similar to the movies the user has rated. Or said differently, if a user A gives a good rating on movie X and other users gives a good rating on movie X and movie Y, then we can recommend the movie Y to the user A.

Mahout computes the recommendations by running several Hadoop mapreduce jobs.
After 30-50 minutes, the jobs are finished and each user will have the 10 movies that she might mostly like based on the co-occurrence of each movie in users' reviews.

To copy and merge the files from HDFS to your local filesystem, type:

```
hadoop fs -getmerge output output.txt
```

The file output.txt should contains lines like this:

```
1       [845:5.0,550:5.0,546:5.0,25:5.0,531:5.0,529:5.0,527:5.0,
2       [546:5.0,288:5.0,11:5.0,25:5.0,531:5.0,527:5.0,515:5.0,5
3       [137:5.0,284:5.0,508:4.8327274,248:4.826923,285:4.80597,
4       [748:5.0,1296:5.0,546:5.0,568:5.0,538:5.0,508:5.0,483:5.
5       [732:5.0,550:5.0,9:5.0,546:5.0,11:5.0,527:5.0,523:5.0,51
6       [739:5.0,9:5.0,546:5.0,11:5.0,25:5.0,531:5.0,528:5.0,527
7       [879:5.0,845:5.0,751:5.0,750:5.0,748:5.0,746:5.0,742:5.0
8       [742:5.0,550:5.0,546:5.0,566:5.0,568:5.0,527:5.0,31:5.0,
9       [739:5.0,550:5.0,546:5.0,11:5.0,527:5.0,523:5.0,514:5.0,
10      [732:5.0,9:5.0,546:5.0,11:5.0,25:5.0,529:5.0,528:5.0,527
```

Each line represents the recommendation for a user. The first number is the user id and the 10 number pairs represents a movie id and a score.
If we are looking at the first line for example, it means that for the user 1, the 10 best recommendations are for the movies 845, 550, 546, 25 ,531, 529, 527, 31, 515, 514.

It's not easy to see what those recommendation means so we wrote a small python program to show for a given user, the movies he has rated and the movies we recommend him.
The python program uses the file u.data for the list of rated movies, the file u.item to get the movie titles and output.txt to get the list of recommended movies for the user.

```
import sys

if len(sys.argv) != 5:
        print "Arguments: userId userDataFilename movieFilename
        sys.exit(1)

userId, userDataFilename, movieFilename, recommendationFilename

print "Reading Movies Descriptions"
movieFile = open(movieFilename)
movieById = {}
for line in movieFile:
        tokens = line.split("|")
        movieById[tokens[0]] = tokens[1:]
movieFile.close()

print "Reading Rated Movies"
userDataFile = open(userDataFilename)
ratedMovieIds = []
for line in userDataFile:
        tokens = line.split("\t")
        if tokens[0] == userId:
                ratedMovieIds.append((tokens[1],tokens[2]))
userDataFile.close()

print "Reading Recommendations"
recommendationFile = open(recommendationFilename)
recommendations = []
for line in recommendationFile:
        tokens = line.split("\t")
        if tokens[0] == userId:
                movieIdAndScores = tokens[1].strip("[]\n").split
                recommendations = [ movieIdAndScore.split(":") f
                break
recommendationFile.close()

print "Rated Movies"
print "------------------------"
for movieId, rating in ratedMovieIds:
        print "%s, rating=%s" % (movieById[movieId][0], rating)
print "------------------------"

print "Recommended Movies"
print "------------------------"
for movieId, score in recommendations:
        print "%s, score=%s" % (movieById[movieId][0], score)
print "------------------------"
```

To run the python program to get the recommended movies for the user 4:

```
$ python show_recommendations.py 4 u.data u.item output.txt
Reading Movies Descriptions
Reading Rated Movies
Reading Recommendations
Rated Movies
------------------------
Mimic (1997), rating=3
Ulee's Gold (1997), rating=5
Incognito (1997), rating=5
One Flew Over the Cuckoo's Nest (1975), rating=4
Event Horizon (1997), rating=4
Client, The (1994), rating=3
Liar Liar (1997), rating=5
Scream (1996), rating=4
Star Wars (1977), rating=5
Wedding Singer, The (1998), rating=5
Starship Troopers (1997), rating=4
Air Force One (1997), rating=5
Conspiracy Theory (1997), rating=3
Contact (1997), rating=5
Indiana Jones and the Last Crusade (1989), rating=3
Desperate Measures (1998), rating=5
Seven (Se7en) (1995), rating=4
Cop Land (1997), rating=5
Lost Highway (1997), rating=5
Assignment, The (1997), rating=5
Blues Brothers 2000 (1998), rating=5
Spawn (1997), rating=2
Wonderland (1997), rating=5
In & Out (1997), rating=5
------------------------
Recommended Movies
------------------------
Saint, The (1997), score=5.0
Indian Summer (1996), score=5.0
Broken Arrow (1996), score=5.0
Speed (1994), score=5.0
Anastasia (1997), score=5.0
People vs. Larry Flynt, The (1996), score=5.0
Casablanca (1942), score=5.0
Trainspotting (1996), score=5.0
Courage Under Fire (1996), score=5.0
Money Talks (1997), score=5.0
------------------------
```

We showed in this tutorial how to use the Mahout recommendation engine. However we only scratched the surface of Mahout capabilities. Indeed, there is a lot more to it and you can see the list of algorithms implemented by Mahout on the Mahout wiki page (https://cwiki.apache.org/MAHOUT/algorithms.html).

FILED UNDER HADOOP TAGGED WITH APACHE, HADOOP, MACHINE LEARNING, MAHOUT, RECOMMENDER

About chimpler
http://www.chimpler.com

27 Responses to *Playing with the Mahout recommendation engine on a Hadoop cluster*

Jason Gowans says:
2013/02/23 at 5:17 pm
Thanks for sharing – really useful.

Reply

chimpler says:
2013/02/24 at 10:24 pm
Thank you Jason!

Reply

Pingback: Using the Mahout Naive Bayes Classifier to automatically classify Twitter messages | Chimpler

Pingback: Generating EigenFaces with Mahout SVD to recognize person faces | Chimpler

Anilabh Pandey says:
2013/04/23 at 8:45 am
that was one of the few mahout/hadoop tutorials that work perfectly. real help for first time experimentation! thanks!

Reply

chimpler says:
2013/04/23 at 8:48 am
Thank you Anilabh!

Reply

Anilabh Pandey says:
2013/05/02 at 1:43 am
is there a way to process incremental data on mahout? so i got 100,000 movies rated and recommended. now i got a new set of 50 movies. some of those have been rated by users. how do i add these new movie recommendations for existing users? do i need to reprocess the entire set (including the new movies) again?

Pingback: Playing with the Mahout recommendation engine o...

Pingback: Finding association rules with Mahout Frequent Pattern Mining | Chimpler

Tarun Gulyani says:
2013/05/13 at 12:26 am
Really Nice Article…Thanks Chimpler…How can we predict rating from user to movie ..instead of recommend the movie to user?

Reply

Renuka SEO says:
2013/07/24 at 2:59 am
The Information which you provided is very much useful for Hadoop Online Training Learners Thank You for Sharing Valuable Information

Reply

rajnishkumargarg says:
2013/08/05 at 5:06 pm
Reblogged this on techtogive and commented:
A really working tutorial on "How to use Mahout recommendation"

Reply

Lays says:
2013/10/10 at 8:39 pm
I configured the JAVA_HOME but I got the following error when I try to format the namenode

JAVA_HOME is not set.

what could it be?

Reply

Lays says:
2013/10/10 at 8:50 pm
sorry, already found what was wrong.

Thank you!

Reply

Pingback: Using Amazon's Elastic Map Reduce to compute recommendations with Apache Mahout 0.8 | Blog of Adam Warski

Pingback: Using Amazon's Elastic MapReduce to Compute Recommendations with Apache Mahout 0.8 | Big Data NewsBig Data News

sudheer1313 says:
2013/11/09 at 4:29 am
Thanks for this valuble information and itis useful for us .Biginfosys also provides the best online Hadoop training classes.

Reply
Pingback: Confluence: Recommendation Engine

Harshit says:
2014/02/26 at 5:19 am
Hi,
I am running recommendation system on a single node hadoop using mahout. It is run on movie data obtained from grouplens (100k data).
Versions:
hadoop version – 1.1.1
mahout-distribution-0.9

I am executing the following command –

hadoop jar /home/avatar/Desktop/Dissertation/Mahout/mahout-distribution-0.9/mahout-core-0.9-job.jar org.apache.mahout.cf.taste.hadoop.item.RecommenderJob -s SIMILARITY_COOCCURRENCE –input /user/hduser/mahout/u.data –output /user/hduser/mahout/output

After a few successful mapreduce tasks, the following error is thrown by each job-
14/02/26 15:10:48 INFO mapred.JobClient: Task Id : attempt_201402261501_0007_m_000000_0, Status : FAILED
Error: org.apache.lucene.util.PriorityQueue.(I)V

What does this error mean, and how to get over with it?
Thanks in advance!

Reply
Pingback: Playing with the Mahout recommendation engine o...

tuanfman says:
2014/08/18 at 5:03 am
Hey guy!
I'm have a bug: when I run " hadoop fs -put u.data u.data" then receive bug: " put: `u.data': No such file or directory " . why ?

Reply
Pingback: Generating EigenFaces with Mahout SVD to recognize person faces | Developer tips & tricks

Supri Amir says:
2014/09/03 at 4:34 am
Hi guys,

How to edit this?
Now set the environment variables:
export HADOOP_PREFIX=
export HADOOP_CONF_DIR=$HADOOP_PREFIX/conf
export PATH=$HADOOP_PREFIX/bin:$PATH

Reply
Supri Amir says:
2014/09/03 at 4:48 am
I am working under Macintosh. So i have trouble to install it

Reply
Pingback: ps3 games

Pingback: Playing with the Mahout recommendation engine on a Hadoop cluster | Big Enterprise Data

Jonathan Napitupulu says:
2015/04/27 at 3:33 am
Thank you. That's very useful tutorial

Reply