Text Clustering Using LucidWorks and Apache Mahout

(Nov. 17, 2012)

1. Module name

Text Clustering Using Lucidworks and Apache Mahout

2. Scope

This module introduces algorithms and evaluation metrics for flat clustering. We focus on the usage of LucidWorks big data analysis software and Apache Mahout, an open source machine learning library in clustering of document collections with the k-means algorithm.

3. Learning objectives

After finishing the exercises, students should be able to

1. Explain the basic idea of k-means and model-based clustering algorithms
2. Explain and apply the k-means algorithm on data collections
3. Evaluate clustering result based on a gold standard set of classes
4. Perform k-means clustering using LucidWorks
5. Perform k-means clustering using Apache Mahout on text collections (Optional)

4. 5S characteristics of the module (streams, structures, spaces, scenarios, society)

1. **Streams:** The input stream to clustering algorithms consists of data vectors. Specifically for text clustering, the input stream consists of tokenized and parsed documents, represented in vectors.
2. **Structures:** Text clustering deals with text collections. Apache Mahout further preprocesses the text collections into a sequence file format.
3. **Spaces:** The indexed documents are converted into vector space for clustering. The document collections are stored on the machine running LucidWorks or Apache Mahout.

4. **Scenarios:** When user want to perform clustering on document collections. That includes getting insight from collections and speeding up nearest-neighbor search algorithms.
5. **Society:** Potential audience includes search engine developers, librarians and data mining specialists.

5. Level of effort required (in-class and out-of-class time required for students)

In-class: 1 hour for lectures and Q&A sessions.

Out-of-class: 3 hours for reading and exercises.

6. Relationships with other modules (flow between modules)

This module is related with module “Text Classification using Mahout”, which talks about using Apache Mahout to perform classification. It also introduces how to install and some basics about Apache Mahout and discusses generating TFIDF vectors.

This module is also related with module “Overview of LucidWorks Big Data software”, which introduces basics about the LucidWorks.

7. Prerequisite knowledge/skills required

1. Basic probability theory.
2. Knowledge about some UNIX shell features.

8. Introductory remedial instruction

8.1 About bash

Bash is a UNIX shell written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell (sh).It is widely used as the default shell on Linux, Mac OS X and Cygwin.

8.2 Environment variables

By command

```
foo=bar
```

we define an environment variable foo to be a string “bar”. We can then use variable foo anywhere by

```
$foo
```

and bash will replace it with bar.

If we run a program in bash, the program will have access to all environment variables in bash. Some programs make use environment variables to define their home directory, so they can access files that are necessary.

There are a number of default environment variables. An important one is HOME. It points to the home directory of the user by default.

8.3 Some special characters

If you want to turn a long command into multiple lines, you can add ‘\’ character at the end of each line, telling bash to go on and read the next line.

In bash, character ‘*’ means “any”. A command

```
rm foo*
```

means remove (delete) everything starting with “foo”, like “foobar.h” or “foo.bar”. With that we can perform some bundle operations.

8.4 Redirection

Bash supports I/O redirection, which allows saving the output of a program for further use, or let the program read inputs from a file. The corresponding characters are ‘>’ and ‘<’.

By command

```
ls > a.txt
```

stdout of ls will be redirected to a.txt. Redirecting stdin using ‘<’ works in the same way.

8.4 Shell script

We can put many commands line by line into a text file, and run that text file in bash, known as shell script.

```
./<name of the text file>
```

The text file should be marked to be executable by command

```
chmod +x <name of the text file>
```

ahead of time. Or an error message may pop up as

```
-bash: ./foo.bar: Permission denied
```

9. Body of knowledge

9.1 K-means

9.1.1 Idea

K-means tries to minimize the average squared Euclidean distance of documents from cluster centroids.

The cluster centroid $\vec{\mu}$of cluster $\omega$is defined as

$$\vec{\mu}(\omega) = \frac{1}{|\omega|}\sum_{\vec{x}\in\omega}\vec{x}$$

Let K be the number of clusters, ω_k as the set of documents in the k[th] cluster, and $\vec{\mu}(\omega_k)$ represents the centroid of the k[th] cluster, k-means tries to minimize

$$\min \text{RSS} = \sum_{k=1}^{K}\sum_{\vec{x}\in\omega}|\vec{x} - \vec{\mu}(\omega_k)|^2$$

9.1.2 Algorithm

1. Select initialize centroids.
2. Reassignment: assign each document vector to its closest centroid in Euclidean distance.
3. Recomputation: update the centroids using the definition of centroid.
4. Loop back to step 2 until stopping criterion is met.

9.1.3 Convergence

K-means is guaranteed to converge because

1. RSS monotonically decreases in each iteration
2. The number of possible cluster assignments is finite, so a monotonically decreasing algorithm will eventually arrive at a (local) minimum

9.1.4 Time complexity

Let K be the number of clusters, N be the number of documents, M be the length of each document vector, and I be the number of iterations.

The time complexity of each iteration: O(KNM)

The time complexity of k-means with a maximum number of iterations: O(IKNM)

9.1.5 Determining cardinality

Cardinality is the number of clusters in data.

We can use the following method to estimate the cardinality in k-means.

1. The "knee" point of the estimated $\mathrm{RSS}_{\min}(\mathrm{K})$ curve. $\mathrm{RSS}_{\min}(\mathrm{K})$ is the minimal RSS of all clusterings with K clusters.
2. The AIC for k-means.

$$\mathrm{AIC}: \quad \mathrm{K} = \arg\min_{\mathrm{K}} [\mathrm{RSS}_{\min}(\mathrm{K}) + 2\mathrm{MK}]$$

M is the length of one document vector.

9.2 Model-based clustering and the Expectation Maximization algorithm

9.2.1 Idea

Model-based clustering assumes that the data were generated by a model, and tries to recover the original model from the data. The model then defines clusters and an assignment of documents is generated along the way.

Maximum likelihood is the criterion often used for model parameters.

$$\Theta = \arg\max_{\Theta} \mathrm{L}(\mathrm{D}|\Theta) = \arg\max_{\Theta} \sum_{\mathrm{n}=1}^{\mathrm{N}} \log \mathrm{P}(\mathrm{d_n}|\Theta)$$

Θ is the set of model parameters, and $\mathrm{D} = \{\mathrm{d}_1, \dots, \mathrm{d}_\mathrm{N}\}$ is the set of document vectors.

This equation means finding a set of model parameters Θ which has the maximum likelihood, or say, the one that gives the maximum log probability to generate the data.

Expectation Maximization or EM algorithm is often used in finding the set of model parameters Θ.

9.3 Evaluation of clustering algorithms

9.3.1 Purity

Given a gold standard set of classes $\mathbb{C} = \{\mathrm{c}_1, \mathrm{c}_2, \dots, \mathrm{c}_\mathrm{j}\}$ and the set of clusters $\Omega = \{\omega_1, \omega_2, \dots, \omega_\mathrm{k}\}$ purity measures how pure the clusters are:

$$\mathrm{purity}(\Omega, \mathbb{C}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{k}} \max |\omega_\mathrm{k} \cap \mathrm{c_j}|$$

9.3.2 Rand index

With N documents in the collection, we can make N(N-1)/2 pairs out of them. We define

	Relationship in gold standard set	Relationship in the set of clusters
True positive (TP)	Same class	Same cluster
True negative (TN)	Different class	Different cluster
False positive (FP)	Different class	Same cluster
False negative (FN)	Same class	Different cluster

Rand index (RI) measures the percentage of decisions that are correct.

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

9.4 Workflow of k-means in LucidWorks

9.4.1 Collection, input directory, etc

In this module we deal with an existing collection “kmeans_reuters”. NOT what we used to use “test_collection_vt”

The input text files are located at hdfs://128.173.49.66:50001/input/reuters/*/*.txt

9.4.2 Submitting a k-means job

With command

```
curl -u username:password -X POST -H 'Content-type:
application/json' -d
'{"doKMeans":"true","kmeans_numClusters":"20","inputDir":
"hdfs://128.173.49.66:50001/input/reuters/*/*.txt","input
Type":"text/plain","collection":"kmeans_reuters"}'
http://fetcher.dlib.vt.edu:8341/sda/v1/client/workflows/_
etl
```

a k-means job will be submitted to the LucidWorks _etl workflow. The "doKMeans":"true" specifies that we are doing k-means. The “inputDir” and “inputType” parameters point the job to the documents we want to deal with. The “collection” parameter specifies the collection where the clustered documents will be stored into. And "kmeans_numClusters":"20" here specifies an optional parameter of number of clusters. A k-means job has the following list of parameters

Name	Description
kmeans_convergenceDelta	Used to determine the point of convergence of the clusters. The default is 0.5.
kmeans_distanceMeasure	Defines the DistanceMeasure class name to use for the clustering. The default is

	org.apache.mahout.common.distance.CosineDistanceMeasure.
kmeans_maxIter	Defines the maximum iterations to run, independent of of the convergence specified. The default is 10.
kmeans_numClusters	Defines the number of clusters to generate. The default is 20.

The k-means job will take quite some time to complete, so the command will return a json file indicating the job information, so we will be able to keep track of the job. For example:

```
{"id":"0000262-120928210911692-oozie-hado-W","workflowId"
:"_etl","createTime":1352997183000,"status":"RUNNING","ch
ildren":[],"throwable":null}
```

We can use command

```
curl -u username:password -X GET
http://fetcher.dlib.vt.edu:8341/sda/v1/client/jobs |
python -mjson.tool
```

to query the status of jobs. Because the list of jobs is long, you can redirect it to a text file for easy reading.

9.4.3 Retrieving clustering results

Upon success, the k-means job updates the cluster assignment of documents in the "clusterId" field in the collection we specified, so we can simply browse the collection and read the clusterId field.

We can either make a json query to kmeans_reuters like

```
curl -u username:password -X POST -H 'Content-type:
application/json' -d '{"query":{"q":"clusterId:*",
"fl":"clusterId,text"}}'
http://fetcher.dlib.vt.edu:8341/sda/v1/client/collections
/kmeans_reuters/documents/retrieval | python -mjson.tool
```

or we can also use the Apache Solr web interface http://fetcher.dlib.vt.edu:8888/solr/#/kmeans_reuters to do this.

9.5 Workflow of k-means in Mahout (Optional)

Here we use Mahout, a machine learning library, to execute k-means algorithm. The k-means algorithm asks for vector input, so we need to get vector representation of the data before we run the algorithm.

9.5.1 Environment settings

As shown in module "Text Classification using Mahout", login to the server using SSH command, the username and password are as follows:

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

Add this command before we start our work:

```
HADOOP_HOME=/home/CS5604/hadoop-0.20.205.0/bin
```

9.5.2 Collection to vectors

To use the k-means algorithm in Mahout, we first need to create sequence file from original data:

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

-i Input data file/directory.

-o Output sequence file/directory.

Then we convert this sequence file to vector:

```
$./bin/mahout seq2sparse -i 20news-seq -o 20news-vectors -lnorm -nv -wttfidf
```

-lnorm If set, the output vectors will be log normalized.

-nv If set, the output vectors will be named vectors.

-wt The weight we are using, can be tf or tfidf.

9.5.3 Choosing initial centroids

Because initialize centroids randomly often impairs performance, Mahout provides methods to select initial centroids for performance. The following command will help us find a good set of initial centroids for k-means:

```
$./bin/mahout canopy -i 20news-vectors/tfidf-vectors -o 20news-centroids
-dm org.apache.mahout.common.distance.SquaredEuclideanDistanceMeasure -t1
500 -t2 250
```

-dm Distance measure. Here we use Euclidean distance Measurement.

All points that lie within the distance t2 to a cluster centroid will not be considered a cluster centroid. These points having distance between t2-t1 to a cluster centroid are like gray areas which can be overlapped by other clusters. These 2 parameters decide how many initial

centroid you'll get from this command and we need to decide them to meet our requirement.

After creating these centroids, in the kmeans-clusters directory we can see a final cluster ending with the word “final”, say“clusters-10-final”, Use cluster-dump utility to check the initial centroid result:

```
$./bin/mahout clusterdump -dtsequencefile -d
20news-vectors/dictionary.file-* -ikmeans-clusters/clusters-10-final -b
10 -n 10 -o report.txt
```

-d Dictionary file in the input data, the dictionary.file- found in 20news-vectors directory.
-i Input.
-o Output of this command, where we can see details about these initial centroids.

9.5.4 Running k-means

With initial centroids and vector inputs, let's run our k-means algorithm as follows:

```
$./bin/mahout kmeans -i 20news-vectors/tfidf-vectors -c 20news-centroids
-o kmeans-clusters -dm
org.apache.mahout.common.distance.SquaredEuclideanDistanceMeasure -cd 1.0
-x 20 -cl
```

-i Input file.
-c Set of initial centroids to start with, which is only needed when the -k parameter is not set. It expects a SequenceFile full of centroids. If the -k parameter is specified, it will erase the folder and write randomly selected k points to a SequenceFile there.
-o Output file.
-dm Distance measure. Here we use Euclidean distance Measurement.
-cd Convergence threshold.
-x Maximum iteration number.
-k The number of clustering centroids (not shown here)

The program will then go many iterations till converge.

Use the cluster-dump command again to see our clustering result. This time use the k-means cluster result in kmeans-clusters directory instead of 20news-centroids directory as the input.

10. Resources

[1]. Owen, S., Anil, R., Dunning, T., and Friedman, E. (2011).Part 2: Clustering. *Mahout in action*. Manning Publications Co..

[2]. Manning, C., Raghavan, P., and Schutze, H. (2008). Chapter 16: Flat Clustering. In *Introduction to Information Retrieval*. Cambridge: Cambridge University Press

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

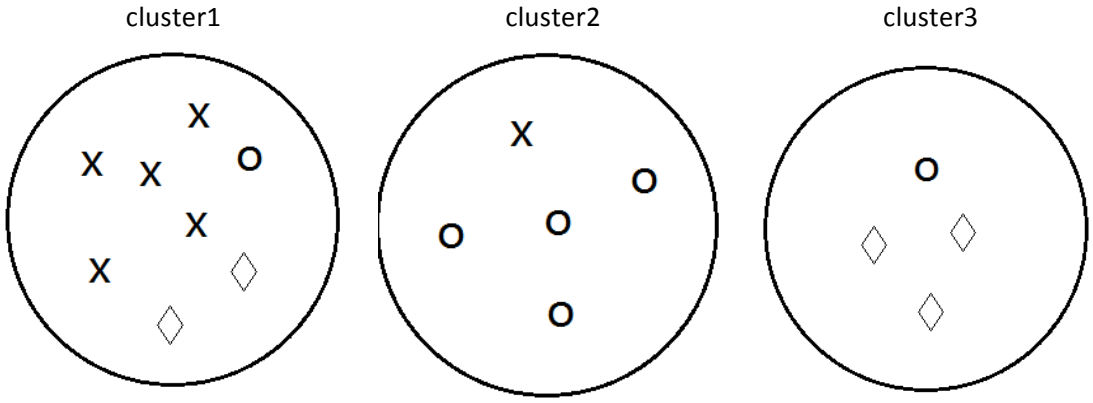
[4]. Job management in LucidWorks: http://lucidworks.lucidimagination.com/display/bigdata/Jobs

[5]. Discovery in LucidWorks: http://lucidworks.lucidimagination.com/display/bigdata/Discovery

11. Exercises / Learning activities

1. Cluster evaluation

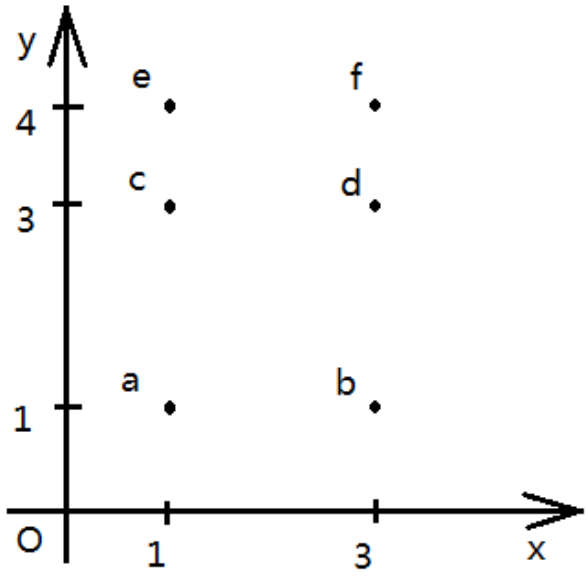
Suppose the diagram below shows the cluster assignment from a clustering algorithm on 17 documents. In the gold standard set of three classes, all crosses belong to one class, all circles belong to one class, and all diamonds belong to one class.



a) Calculate the purity measure and the rand index measure of this clustering.

b) Replace every document d in the above diagram with two identical copies of d in the same class, calculate the purity measure and the rand index measure on the clustering with 34 points, and describe the changes on the two measures compared with the result in a).

2. K-means



Assume we have 6 data vectors

a	(1,1)
b	(3,1)
c	(1,3)
d	(3,3)
e	(1,4)
f	(3,4)

They are shown in the above graph. Now we want to use k-means algorithm to cluster those points into K=2 clusters.

Assume we initialize the centroids as $\mu_1 = e,\ \mu_2 = d$. The algorithm stops when RSS stops decreasing, cluster assignment stops changing or the number of iterations reaches 8.

a) Perform the k-means algorithm, list all intermediate result like centroids and assignments of the reassignment and recalculation steps, and show the final cluster assignment.

b) For the cluster assignment from a), discuss if we obtained the global minimum of RSS or not. That is, does the cluster assignment in a) give minimal RSS?

3. Practice k-means using LucidWorks

a) Submit a k-means job to the LucidWorks server with the existing target collection "kmeans_reuters", input directory hdfs://128.173.49.66:50001/input/reuters/*/*.txt and input type text/plain, of 10 clusters. Show the command you use, and the job ID returned.

b) Make queries on job status to keep track of your job. Wait till your job complete. In your answer, show the succeed status.

c) Retrieve the clustered documents, and generate word clouds with the "text" field of documents in each cluster. In your answer, show word clouds of at least 3 of the clusters.

There are 122 documents in total. If you find only 10 documents are retrieved, set a larger "rows" parameter in your query.

Here a hint with web interface is that you can ask the query to return in csv format, so you can make a text file, paste everything into it, change its affix to csv, and open it in Excel for easy processing.

Word cloud visualizes text in a "cloud" of words. For this exercise, you can simply paste everything in the text field of documents in one cluster into a free online tool www.wordle.net/create and hit "go" button to create a word cloud.

12. Evaluation of learning objective achievement

The completion of learning objective shall be evaluated by the correctness and level of understanding in students' response to the exercises.

13. Glossary

14. Additional useful links

Wikipedia: Bash (Unix shell): http://en.wikipedia.org/wiki/Bash_(Unix_shell)

Wikipedia: Environment variable: http://en.wikipedia.org/wiki/Environment_variable

Wikipedia: Shell script: http://en.wikipedia.org/wiki/Shell_script

Wikipedia : Redirection: http://en.wikipedia.org/wiki/Redirection_(computing)

15. Contributors

Authors: Liangzhe Chen (liangzhe@vt.edu), Xiao Lin(linxiao@vt.edu) and Andrew Wood(andywood@vt.edu)

Reviewers: Dr. Edward A. Fox, Kiran Chitturi, Tarek Kanan

Class: CS 5604: Information Retrieval and Storage. Virginia Polytechnic Institute and State University, Fall 2012.