Getting Started On KMeans Clustering

1. Create a new directory on your machine called "KMeans".

2. Download "KMeans.zip" from here. Put it into the KMeans directory and unzip it. Among other things, this will create a "20_Newsgroups" directory that contains the "20Newsgroups" data set. This is a set of around 20,000 postings to 20 different newsgroups (in case you are too young to have experienced newsgroups, these were sort of like the blogs of the 1990s). The "20_newsgroups" directory will have 20 subdirectories, each corresponding to a newsgroup. There are around 1000 files in each subdirectory, each of which is a posting.

4. Inside of the KMeans directory, you will also have the files "ProcessCorpus.jar", "GetCentroids.jar", "KMeans.jar", and "GetDistribution.jar". These are compiled Java archive. First we will run "ProcessCorpus.jar", which will take all of the newsgroup postings and turn them into bag-of-words vectors. Run it using:

```
java -jar ProcessCorpus.jar
```

When prompted, answer:

```
Enter the directory where the corpus is located: 20_newsgroups
Enter the name of the file to write the result to: vectors
Enter the max number of docs to use in each subdirectory: 100
How many of those words do you want to use...? 10000
```

This will create a file called "vectors" that has 20 * 100 lines, each of which is a vectorized representation of a document. The dictionary size that is used is 10000 words.

5. Now, you'll run a program that will choose an initial set of centroids from the data:

```
java -jar GetCentroids.jar
```

When prompted, answer:

```
Enter the data file to select the clusters from: vectors
Enter the name of the file to write the result to: clusters
Enter the number of clusters to select: 20
```

6. This will create a file called "clusters" that has 20 lines, each of which describes a cluster centroids (you might make a copy of this file, calling it “clustersCopy” so that you can use it when you complete step 8 below). This initial set of cluster centroids is simply randomly sampled from the "vectors" file. You can now cluster the data using:

```
java -jar KMeans.jar
```

Then when prompted:

```
Enter the file with the data vectors: vectors
Enter the name of the file where the clusters...: clusters
Enter the number of iterations to run: 10
```

It might take five minutes to run (in the meantime, you can go to step 7 below). When it completes, you can examine the results by running:

```
java -jar GetDistribution.jar
```

Then when prompted:

```
Enter the file with the data vectors: vectors
Enter the name of the file where the clusters...: clusters
```

This will go off and count how many of each of the 20 types of newsgroups were put into each cluster. If the clustering where ineffective, you would expect an even distribution of newsgroups into clusters. But what you should find it that the clusters tend to group the postings together in meaningful (and interesting!) ways.

7. You'll also notice that a "SeqKMeans" directory was created when you unzipped "KMeans.zip". If you look in there, you'll see a bunch of files that are used to implement the KMeans clustering algorithm. Only

four are important to you. Take a look at them:

(a) "IDoubleVector.java" and "SparseDoubleVector.java". The former is an interface that provides a bunch of operations on vectors of Double values (adding vectors, subtracting them, finding the distance between them, and so on). The latter file provides a sparse implementation of IDoubleVector.

(b) "VectorizedObject,java". This is the object that is used to store both the vectorized documents, and the clusters. In the case of a vectorized document, the "key" is the name of the document, and the "value" is the name of the newsgroup that the document is in. In the case of a cluster, the "key" is the name of the cluster, and the "value" is the number of documents that have been assigned to the cluster.

(c) "KMeans.java". This is the actual clustering code.

8. Your task is to fill in the 20-30 lines of code that are missing from "KMeans.java", compile all of these Java files, and then run your code to see if you can cluster the data!

If you decide that you want a little help, you can see the full KMeans.java file (without any missing code) here.