DATSCIW261 ASSIGNMENT #4

Angela Gunn, Jing Xu

angela@egunn.com, jaling@gmail.com

W261-3

DATSCIW261 Assignment #4

2/10/16

HW4.0

What is MrJob?

MRJob is a Python package for running Hadoop streaming jobs developed by Yelp in 2010. The motivation for its development was a desire for a better parallel solution across multiple machines. The goals for the framework are:

1. Simplicity and a pseudo-code syntax
2. Flexibility and generality
3. Easy installation
4. Performance

MRJob assists in producing data pipelines and multistep jobs that feed once MapReduce output directly as input into the next MapReduce job, and then submitting the multistep job to Hadoop job tracker. Some other features are an abstracted MapReduce Interface, the ability to handle complex Python objects, acceptance of a variety of input formats (text based, binary with pickled objects), AWS integration, easy to run locally, and a strong community of developers and contributors.

How is it different to Hadoop MapReduce?

Hadoop MapReduce is a framework to process large data sets with programs running in a distributed and fault tolerant way. Hadoop MapReduce is just one framework that MRJob can run on. MrJob is a python wrapper over hadoop streaming API to provide a consistent interface to run programs on a variety of environments, be it local, hadoop cluster or AWS EMR without changing the code.

What are the mapper_init(), mapper_final(), combiner_final(), reducer_final() methods? When are they called?

mapper_init() is used to define the actions needed to set up the mapping algorithm, and is the called before the mapper function processes any input. This may include loading file information needed for the mapping step. It runs before the mapper function.
mapper_final() is used to define an action to be run after the mapepr function reaches the end of the inputs.
combiner_final() defines the actions necessary to clean up the combiner algorithms, running after the combiner finishes its execution.
reducer_final() defines the actions necessary to clean up the combiner algorithms, running after the reducer finishes its execution.

HW4.1

What is serialization in the context of MrJob or Hadoop?

Serialization is the process of converting an object into a stream of bytes in order to transmit it over a network into memory for interprocess communication or to persistent storage. The purpose is to save the state of an object until it is needed again. Text processing is slow, and incurs extra storage and networks costs, so serialization is method of more efficient communication and storage. The reverse process is called deserialization.

When it used in these frameworks?

In these frameworks, serialization enables the data to be compacted into binary format so large data sets can be transferred over the network and consumed by various programming languages. It is used between the tasks of a mapreduce job such as map, combine and reduce.

What is the default serialization mode for input and outputs for MrJob?

The default serialization mode in MRJob for input is RawValueProtocol (raw text value). The output is JSONProtocol (in JSON format).

HW4.2

Recall the Microsoft logfiles data from the async lecture. The logfiles are described are located at:

https://kdd.ics.uci.edu/databases/msweb/msweb.html (https://kdd.ics.uci.edu/databases/msweb/msweb.html) http://archive.ics.uci.edu/ml/machine-learning-databases/anonymous/ (http://archive.ics.uci.edu/ml/machine-learning-databases/anonymous/)

This dataset records which areas (Vroots) of www.microsoft.com each user visited in a one-week timeframe in Feburary 1998.

Here, you must preprocess the data on a single node (i.e., not on a cluster of nodes) from the format:

C,"10001",10001 #Visitor id 10001

V,1000,1 #Visit by Visitor 10001 to page id 1000

V,1001,1 #Visit by Visitor 10001 to page id 1001

V,1002,1 #Visit by Visitor 10001 to page id 1002

C,"10002",10002 #Visitor id 10001

V

Note: #denotes comments to the format:

V,1000,1,C, 10001

V,1001,1,C, 10001

V,1002,1,C, 10001

Write the python code to accomplish this

In [1]:
```
%%writefile log_preprocess_42.py
#!/usr/bin/python
## log_preprocess_42.py
## Author: Angela Gunn & Jing Xu
## Description: Proprocesses log data on a single node
import sys
import os

if len(sys.argv) < 2:
    print "No input file is passed, Aborting!!!"
    sys.exit(1)

input_file = sys.argv[1]
output_file = input_file + '.pp'

try:
    os.remove(output_file)
except OSError:
    pass

last_visitor = None #set last visitor value to append to output fil
e
with open(input_file, 'r') as f1: #open input file to read
    with open(output_file, 'a') as f2: #open ouput file to write
        for line in f1:
            line = line.strip()
            tokens = line.split(",")
            if len(tokens) == 3 and tokens[0] == 'C': #check for Vi
sitor ID
                last_visitor = tokens[2]  #set last visitor to new
Visitor ID
            if len(tokens) == 3 and tokens[0] == 'V': #check for Pa
ge ID
                out_line = 'V,{0},C, {1}\n'.format(tokens[1],last_v
isitor)
                f2.write(out_line)
```

```
Overwriting log_preprocess_42.py
```

In [2]:
```
!chmod a+x log_preprocess_42.py
```

In [3]:
```
!python log_preprocess_42.py anonymous-msweb.data
```

In [4]:
```
!echo "first 10 lines"
!head -n 10 anonymous-msweb.data.pp
!echo "last 10 lines"
!tail -n 10 anonymous-msweb.data.pp
```

```
first 10 lines
V,1000,C, 10001
V,1001,C, 10001
V,1002,C, 10001
V,1001,C, 10002
V,1003,C, 10002
V,1001,C, 10003
V,1003,C, 10003
V,1004,C, 10003
V,1005,C, 10004
V,1006,C, 10005
last 10 lines
V,1123,C, 42708
V,1038,C, 42708
V,1026,C, 42708
V,1041,C, 42708
V,1001,C, 42709
V,1003,C, 42709
V,1035,C, 42710
V,1001,C, 42710
V,1018,C, 42710
V,1008,C, 42711
```

HW 4.3

Find the 5 most frequently visited pages using MrJob from the output of 4.2 (i.e., transfromed log file)

In [5]:

```
%%writefile top_pages_43.py
## top_pages_43.py
## Author: Angela Gunn & Jing Xu
## Description: Find 5 most frequently visited pages from the log

from mrjob.job import MRJob
from mrjob.step import MRStep
import csv

def csv_readline(line):
    """Given a sting CSV line, return a list of strings."""
    for row in csv.reader([line]):
        return row

class TopPages(MRJob):

    top5 = {} #initialize top5 dictionary

    def steps(self):
        return [MRStep(mapper = self.mapper,
                       combiner = self.combiner,
                       reducer = self.reducer),
                MRStep(reducer = self.output_find_top_5)]

    def mapper(self, line_no, line):
        #Extracts the Vroot that was visited
        line = line.strip(' ')
        cell = csv_readline(line)
        yield cell[1],1

    def combiner(self, vroot, visit_counts):
        #combines the visits
        total = sum(visit_counts)
        yield vroot, total

    def reducer(self, vroot, visit_counts): #Sumarizes the visit co
unts by adding them together.
        #combines the visits, and adds vroot to top5 dictionary if
qualified
        total = sum(visit_counts)
        if len(self.top5) < 5:            #less than 5 items, so ad
d
            self.top5[vroot] = total
        else:
            #must find the smallest item; if smaller than the new i
tem, delete it and add new item.
            top_min = min(self.top5, key=self.top5.get)
            if total >= self.top5[top_min]:
                del self.top5[top_min]
                self.top5[vroot] = total
        yield vroot, total
    #end def reducer

    def output_find_top_5(self, vroot, visit_counts):
```

```
        #outputs the results of our top 5
        if len(self.top5) > 0:
            top_max = max(self.top5, key=self.top5.get)
            yield top_max, self.top5.pop(top_max)

if __name__ == '__main__':
    TopPages.run()
```

```
Overwriting top_pages_43.py
```

In [6]:
```
!chmod a+x top_pages_43.py
```

In [7]:
```
%reload_ext autoreload
%autoreload 2
from top_pages_43 import TopPages
import csv

mr_job = TopPages(args=['anonymous-msweb.data.pp'])
with mr_job.make_runner() as runner:
    runner.run()
    for line in runner.stream_output():
        print mr_job.parse_output_line(line)
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:

('1008', 10836)
('1034', 9383)
('1004', 8463)
('1018', 5330)
('1017', 5108)
```

HW 4.4

Find the most frequent visitor of each page using MrJob and the output of 4.2 (i.e., transfromed log file). In this output please include the webpage URL, webpageID and Visitor ID.

In [8]:
```
# Create a file with only URL(s), i.e. records starting with 'A'
!rm -v url.txt
!grep ^A anonymous-msweb.data > url.txt
```

```
url.txt
```

In [9]:

```
%%writefile top_visitor_44.py
## top_visitor_44.py
## Author: Angela Gunn & Jing Xu
## Description: Find most frequent visitor for each page from the l
og

from mrjob.job import MRJob
from mrjob.step import MRStep
import csv

def csv_readline(line):
    """Given a sting CSV line, return a list of strings."""
    for row in csv.reader([line]):
        return row

class TopVisitor(MRJob):

    top_page_visitor = {}

    def steps(self):
        return [MRStep(mapper = self.mapper,
                       combiner = self.combiner,
                       reducer = self.reducer),
                MRStep(reducer = self.reducer_frequent_visitor)]

    def mapper(self, line_no, line):
        #Extracts the Vroot that was visited
        line = line.strip(' ')
        cell = csv_readline(line)
        yield (cell[1],cell[3]),1

    def combiner(self, key, visit_counts):
        #combines the visits
        total = sum(visit_counts)
        yield key, total

    def reducer(self, key, visit_counts): #Sumarizes the visit coun
ts by adding them together.
        #combines the visits, and adds the key to top_page_visitor
dictionary if qualified
        total = sum(visit_counts)
        page = key[0]
        visitor = key[1][1:]
        top_count = int(self.top_page_visitor.get(page,(visitor,0))
[1]) #assign top_count value
        if top_count < total:
            self.top_page_visitor[page] = (visitor,total)
        yield page, total
    #end def reducer

    def reducer_frequent_visitor(self, page, visit_counts):
        with open('url.txt','r') as f:
            for line in f:
                cell = csv_readline(line)
```

```
                if cell[1] == page:
                    key = "{0:>20}  |  {1:>5}  |  {2:>5}".format(ce
ll[4],page,self.top_page_visitor[page][0]) #yield top page visitor
                    break
        yield key, self.top_page_visitor[page][1]

if __name__ == '__main__':
    TopVisitor.run()
```

```
Overwriting top_visitor_44.py
```

In [10]:
```
!chmod a+x top_visitors_44.py
```

```
chmod: top_visitors_44.py: No such file or directory
```

In [11]:
```
!echo "Note: listing first user with max visits for each page"
!echo "                 url    page_id   visitor    visits"
!cat output_hw44.txt
```

Note: listing first user with max visits for each page

url	page_id	visitor	visits
/regwiz	1000	10001	1
/support	1001	10001	1
/athome	1002	10001	1
/kb	1003	10002	1
/search	1004	10003	1
/norge	1005	10004	1
/misc	1006	10005	1
/ie_intl	1007	10007	1
/msdownload	1008	10009	1
/windows	1009	10009	1
/vbasic	1010	10010	1
/officedev	1011	10010	1
/outlookdev	1012	10010	1
/vbasicsupport	1013	10010	1
/officefreestuff	1014	10010	1
/msexcel	1015	10011	1
/excel	1016	10011	1
/products	1017	10011	1
/isapi	1018	10011	1
/mspowerpoint	1019	10011	1
/msdn	1020	10012	1
/visualc	1021	10012	1
/truetype	1022	10013	1
/spain	1023	10014	1
/iis	1024	10015	1
/gallery	1025	10016	1
/sitebuilder	1026	10016	1
/intdev	1027	10017	1
/oledev	1028	10017	1
/clipgallerylive	1029	10019	1
/ntserver	1030	10019	1
/msoffice	1031	10019	1
/games	1032	10019	1
/logostore	1033	10019	1
/ie	1034	10020	1
/windowssupport	1035	10021	1
/organizations	1036	10021	1
/windows95	1037	10021	1
/sbnmember	1038	10021	1
/isp	1039	10021	1
/office	1040	10021	1
/workshop	1041	10021	1
/vstudio	1042	10021	1
/smallbiz	1043	10021	1
/mediadev	1044	10024	1
/netmeeting	1045	10025	1
/iesupport	1046	10027	1
/publisher	1048	10030	1
/supportnet	1049	10031	1
/macoffice	1050	10032	1
/scheduleplus	1051	10035	1
/word	1052	10035	1

/visualj	1053	10035	1
/exchange	1054	10036	1
/kids	1055	10037	1
/sports	1056	10037	1
/powerpoint	1057	10038	1
/referral	1058	10039	1
/sverige	1059	10047	1
/msword	1060	10053	1
/promo	1061	10058	1
/msaccess	1062	10065	1
/intranet	1063	10067	1
/activeplatform	1064	10068	1
/java	1065	10068	1
/musicproducer	1066	10068	1
/frontpage	1067	10068	1
/vbscript	1068	10068	1
/windowsce	1069	10068	1
/activex	1070	10068	1
/automap	1071	10068	1
/vinterdev	1072	10068	1
/taiwan	1073	10078	1
/ntworkstation	1074	10085	1
/jobs	1075	10104	1
/ntwkssupport	1076	10107	1
/msofficesupport	1077	10109	1
/ntserversupport	1078	10122	1
/australia	1079	10122	1
/brasil	1080	10126	1
/accessdev	1081	10127	1
/access	1082	10127	1
/msaccesssupport	1083	10127	1
/uk	1084	10132	1
/exchangesupport	1085	10132	1
/oem	1086	10132	1
/proxy	1087	10132	1
/outlook	1088	10132	1
/officereference	1089	10132	1
/gamessupport	1090	10133	1
/hwdev	1091	10142	1
/vfoxpro	1092	10150	1
/vba	1093	10156	1
/mshome	1094	10156	1
/catalog	1095	10156	1
/mspress	1096	10156	1
/latam	1097	10156	1
/devonly	1098	10157	1
/cio	1099	10159	1
/education	1100	10165	1
/oledb	1101	10166	1
/homeessentials	1102	10168	1
/works	1103	10168	1
/hk	1104	10191	1
/france	1105	10197	1
/cze	1106	10198	1

/slovakia	1107	10198	1
/teammanager	1108	10205	1
/technet	1109	10205	1
/mastering	1110	10208	1
/ssafe	1111	10208	1
/canada	1112	10208	1
/security	1113	10215	1
/servad	1114	10216	1
/hun	1115	10216	1
/switzerland	1116	10225	1
/sidewinder	1117	10228	1
/sql	1118	10235	1
/corpinfo	1119	10240	1
/switch	1120	10241	1
/magazine	1121	10241	1
/mindshare	1122	10243	1
/germany	1123	10254	1
/industry	1124	10263	1
/imagecomposer	1125	10269	1
/mediamanager	1126	10272	1
/netshow	1127	10286	1
/msf	1128	10286	1
/ado	1129	10290	1
/syspro	1130	10306	1
/moneyzone	1131	10316	1
/msmoneysupport	1132	10316	1
/frontpagesupport	1133	10319	1
/backoffice	1134	10335	1
/mswordsupport	1135	10339	1
/usa	1136	10348	1
/mscorp	1137	10348	1
/mind	1138	10351	1
/k-12	1139	10362	1
/netherlands	1140	10363	1
/europe	1141	10372	1
/southafrica	1142	10372	1
/workshoop	1143	10381	1
/devnews	1144	10406	1
/vfoxprosupport	1145	10418	1
/msp	1146	10429	1
/msft	1147	10438	1
/channel_resources	1148	10468	1
/adc	1149	10471	1
/infoserv	1150	10473	1
/mspowerpointsupport	1151	10482	1
/rus	1152	10486	1
/venezuela	1153	10500	1
/project	1154	10564	1
/sidewalk	1155	10606	1
/powered	1156	10624	1
/win32dev	1157	10627	1
/imedia	1158	10632	1
/transaction	1159	10661	1
/visualcsupport	1160	10669	1

/workssupport	1161	10677	1
/infoservsupport	1162	10699	1
/opentype	1163	10744	1
/smsmgmt	1164	10752	1
/poland	1165	10784	1
/mexico	1166	10788	1
/hwtest	1167	10791	1
/salesinfo	1168	10797	1
/msproject	1169	10797	1
/mail	1170	10821	1
/merchant	1171	10828	1
/belgium	1172	10834	1
/moli	1173	10842	1
/nz	1174	10866	1
/msprojectsupport	1175	10888	1
/jscript	1176	10932	1
/events	1177	10951	1
/msdownload.	1178	11008	1
/colombia	1179	11027	1
/slovenija	1180	11035	1
/kidssupport	1181	11044	1
/fortran	1182	11090	1
/italy	1183	11111	1
/msexcelsupport	1184	11134	1
/sna	1185	11142	1
/college	1186	11150	1
/odbc	1187	11173	1
/korea	1188	11190	1
/internet	1189	11243	1
/repository	1190	11287	1
/management	1191	11331	1
/visualjsupport	1192	11359	1
/offdevsupport	1193	11367	1
/china	1194	11372	1
/portugal	1195	11429	1
/ie40	1196	11431	1
/sqlsupport	1197	11444	1
/pictureit	1198	11482	1
/feedback	1199	11644	1
/benelux	1200	11674	1
/hardware	1201	11800	1
/advtech	1202	11802	1
/danmark	1203	11806	1
/msscheduleplus	1204	11904	1
/hardwaresupport	1205	11917	1
/select	1206	12011	1
/icp	1207	12135	1
/israel	1208	12177	1
/turkey	1209	12239	1
/snasupport	1210	12359	1
/smsmgmtsupport	1211	12395	1
/worldwide	1212	12421	1
/corporate_solutions	1213	12472	1
/finserv	1214	12515	1

/developer	1215	12577	1
/vrml	1216	12666	1
/ireland	1217	12675	1
/publishersupport	1218	12714	1
/ads	1219	12746	1
/macofficesupport	1220	12795	1
/mstv	1221	12815	1
/msofc	1222	12819	1
/finland	1223	12828	1
/atec	1224	13041	1
/piracy	1225	13061	1
/msschedplussupport	1226	13179	1
/argentina	1227	13235	1
/vtest	1228	13266	1
/uruguay	1229	13510	1
/mailsupport	1230	13541	1
/win32devsupport	1231	13918	1
/standards	1232	13926	1
/vbscripts	1233	14363	1
/off97cat	1234	14418	1
/onlineeval	1235	14522	1
/globaldev	1236	14738	1
/devdays	1237	14764	1
/exceldev	1238	15247	1
/msconsult	1239	15361	1
/thailand	1240	15461	1
/india	1241	15820	1
/msgarden	1242	16289	1
/usability	1243	16904	1
/devwire	1244	16967	1
/ofc	1245	16999	1
/gamesdev	1246	17120	1
/wineguide	1247	18240	1
/softimage	1248	18347	1
/fortransupport	1249	18384	1
/middleeast	1250	18534	1
/referencesupport	1251	18941	1
/giving	1252	19483	1
/worddev	1253	19746	1
/ie3	1254	20190	1
/msmq	1255	20277	1
/sia	1256	20832	1
/devvideos	1257	21184	1
/peru	1258	21399	1
/controls	1259	21424	1
/trial	1260	21894	1
/diyguide	1261	22485	1
/chile	1262	24951	1
/services	1263	26122	1
/se_partners	1264	26801	1
/ssafesupport	1265	26811	1
/licenses	1266	26815	1
/caribbean	1267	27482	1
/javascript	1268	27503	1

/business	1269	28044	1
/developr	1270	28493	1
/mdsn	1271	28493	1
/softlib	1272	28493	1
/mdn	1273	28493	1
/pdc	1274	28493	1
/security.	1275	28903	1
/vtestsupport	1276	29654	1
/stream	1277	30111	1
/hed	1278	30460	1
/msgolf	1279	31062	1
/music	1280	33424	1
/intellimouse	1281	37099	1
/home	1282	39877	1
/cinemania	1283	41033	1
/partner	1284	41108	1
/train_cert	1295	10028	1

HW 4.5

Here you will use a different dataset consisting of word-frequency distributions for 1,000 Twitter users. These Twitter users use language in very different ways, and were classified by hand according to the criteria:

0: Human, where only basic human-human communication is observed.

1: Cyborg, where language is primarily borrowed from other sources (e.g., jobs listings, classifieds postings, advertisements, etc...).

2: Robot, where language is formulaically derived from unrelated sources (e.g., weather/seismology, police/fire event logs, etc...).

3: Spammer, where language is replicated to high multiplicity (e.g., celebrity obsessions, personal promotion, etc...)

Check out the preprints of our recent research, which spawned this dataset:

http://arxiv.org/abs/1505.04342 (http://arxiv.org/abs/1505.04342) http://arxiv.org/abs/1508.01843 (http://arxiv.org/abs/1508.01843)

The main data lie in the accompanying file:

topUsers_Apr-Jul_2014_1000-words.txt

and are of the form:

USERID,CODE,TOTAL,WORD1_COUNT,WORD2_COUNT,... . .

where

USERID = unique user identifier

CODE = 0/1/2/3 class code

TOTAL = sum of the word counts

Using this data, you will implement a 1000-dimensional K-means algorithm in MrJob on the users by their 1000-dimensional word stripes/vectors using several centroid initializations and values of K.

Note that each "point" is a user as represented by 1000 words, and that word-frequency distributions are generally heavy-tailed power-laws (often called Zipf distributions), and are very rare in the larger class of discrete, random distributions. For each user you will have to normalize by its "TOTAL" column. Try several parameterizations and initializations:

(A) K=4 uniform random centroid-distributions over the 1000 words

(B) K=2 perturbation-centroids, randomly perturbed from the aggregated (user-wide) distribution

(C) K=4 perturbation-centroids, randomly perturbed from the aggregated (user-wide) distribution

(D) K=4 "trained" centroids, determined by the sums across the classes.

and iterate until a threshold (try 0.001) is reached. After convergence, print out a summary of the classes present in each cluster. In particular, report the composition as measured by the total portion of each class type (0-3) contained in each cluster, and discuss your findings and any differences in outcomes across parts A-D.

Note that you do not have to compute the aggregated distribution or the class-aggregated distributions, which are rows in the auxiliary file:

topUsers_Apr-Jul_2014_1000-words_summaries.txt

Functionality of MRJob

mapper_init: load latest centroids files before running the mapper

mapper: read input stream and emit key = cluster index and value = tuple(features, class counts)

combiner: read mapper output and combine features for the same cluster and aggregate class counts

reducer: emit new centroid with class counts

In [12]:

```
%%writefile Kmeans_45.py
#!/usr/bin/python
## Kmeans_45.py
## Author: Angela Gunn & Jing Xu
## Description: Does kmeans

import numpy as np
from numpy import argmin, array, random
from mrjob.job import MRJob
from mrjob.step import MRJobStep
from itertools import chain




#Calculate find the nearest centroid for data point
def MinDist(datapoint, centroid_points):
    # calculate euclidean distance
    euclidean_distance = np.sum((datapoint - centroid_points)**2, a
xis = 1)
    # get the nearest centroid for each instance
    minidx = np.argmin(euclidean_distance)
    return minidx

#Check whether centroids converge
def stop_criterion(centroid_points_old, centroid_points_new,T):
    return np.alltrue(abs(np.array(centroid_points_new) - np.array
(centroid_points_old)) <= T)

class MRKmeans(MRJob):
    centroid_points=[]
    CENTROID = "Centroids.txt"
    #k=4
    def steps(self):
        return [
            MRJobStep(mapper_init = self.mapper_init, mapper=self.m
apper,combiner = self.combiner,reducer=self.reducer)
                ]

    #load centroids info from file
    def mapper_init(self):
        self.centroid_points = [map(float,s.split('\n')[0].split
(',')) for s in open(self.CENTROID).readlines()]
        open(self.CENTROID, 'w').close()

    #load data and output the nearest centroid index and data point
    # returns key = nearest centroid, values = tuple(features, clas
s:1)
    def mapper(self, _, line):
        terms = line.strip().split(',')
        userid = terms[0]
        code = int(terms[1]) #what type of user
        total = int(terms[2])
        features = np.array([float(x) / total  for x in terms[3:]])
```

```
#normalize; features = words

        # key    = centroid
        # values = tuple(features, code:1)
        yield int(MinDist(features, self.centroid_points)), (list(f
eatures), {code:1})

    #Combine sum of data points locally
    # returns key = idx, values = tuple(features, class:n) where n
is the new count
    def combiner(self, idx, inputdata):
        combine_features = None  #features = words
        combine_codes = {}       #codes = class

        for features, code in inputdata:  #for each input line, get
the features (word counts) and the class info
            features = np.array(features)

            # local aggregate of features
            if combine_features is None:
                combine_features = np.zeros(features.size)
            combine_features += features

            # count number of codes
            for k, v in code.iteritems():
                combine_codes[k] = combine_codes.get(k, 0) + v

        yield idx, (list(combine_features), combine_codes)

    #Aggregate sum for each cluster and then calculate the new cent
roids
    #same as reducer, but calculates new centroids as key instead o
f feature list.
    def reducer(self, idx, inputdata):
        combine_features = None  #features = words
        combine_codes = {}       #codes = class

        for features, code in inputdata:  #for each input line, get
the features (word counts) and the class info
            features = np.array(features)

            # local aggregate of features
            if combine_features is None:
                combine_features = np.zeros(features.size)
            combine_features += features

            # count number of codes
            for k, v in code.iteritems():
                combine_codes[k] = combine_codes.get(k, 0) + v

        # new centroids
        centroids = combine_features / sum(combine_codes.values())

        yield idx, (list(centroids), combine_codes)
```

```
if __name__ == '__main__':
    MRKmeans.run()
```

```
Overwriting Kmeans_45.py
```

In [13]:
```
!chmod a+x Kmeans_45.py
```

The below code is common for all questions for 4.5. It contains the function kmeans, which takes a value k (clusters), the centroid_points and the ultimate output file.

The centroid_points is determined for each part of this question separately.

In [14]:

```
%reload_ext autoreload
%autoreload 2
from numpy import random
import numpy as np
from Kmeans_45 import MRKmeans, stop_criterion

THRESHOLD = 0.001
CENTROIDS = "Centroids.txt"


def kmeans(k, centroid_points, output_file):

    mr_job = MRKmeans(args=['--file', output_file,
                            '--file', CENTROIDS,
                                '--jobconf', 'k={0}'.format(k),
                                'topUsers_Apr-Jul_2014_1000-words.t
xt', '-v'])


    # Update centroids iteratively
    i = 1
    while(1):
        # save previous centoids to check convergency
        centroid_points_old = centroid_points

        with mr_job.make_runner() as runner:
            #print "running iteration" + str(i) + ":"
            runner.run()
            centroid_points = []
            clusters = {}

            # stream_output: get access of the output
            for line in runner.stream_output():
                key, value =  mr_job.parse_output_line(line)
                centroid, codes = value
                centroid_points.append(centroid)
                clusters[key] = codes

        print "working" + "." * (i)
        if(stop_criterion(centroid_points_old, centroid_points, THR
ESHOLD)):
            #Let's print some output!

            # display statistics
            print "cluster distribution"
            print "-" * 80
            print "iteration # {}".format(i)
            codes = { 0:'Human', 1:'Cyborg', 2:'Robot', 3:'Spammer'
}

            human_total   = np.sum([clusters[k].get('0', 0) for k i
n clusters.keys()])
            cyborg_total  = np.sum([clusters[k].get('1', 0) for k i
n clusters.keys()])
```

```
            robot_total   = np.sum([clusters[k].get('2', 0) for k i
n clusters.keys()])
            spammer_total = np.sum([clusters[k].get('3', 0) for k i
n clusters.keys()])

            print "-" * 80
            max_class = {}
            print "{0:>5} |{1:>12} (%) |{2:>12} (%) |{3:>12} (%) |
{4:>12} (%)".format("k",

str(human_total) + " Human",

str(cyborg_total) + " Cyborg",

str(robot_total) + " Robot",

str(spammer_total) + " Spammer")
            print "-" * 80
            for cluster_id, cluster in clusters.iteritems():
                total = sum(cluster.values())
                print "{0:>5} | {1:>5} ({2:6.2f}%) | {3:>5} ({4:6.2
f}%) | {5:>5} ({6:6.2f}%) | {7:>5} ({8:6.2f}%)".format(
                    cluster_id,
                    cluster.get('0', 0),
                    float(cluster.get('0', 0))/human_total*100,
                    cluster.get('1', 0),
                    float(cluster.get('1', 0))/cyborg_total*100,
                    cluster.get('2', 0),
                    float(cluster.get('2', 0))/robot_total*100,
                    cluster.get('3', 0),
                    float(cluster.get('3', 0))/spammer_total*100
                )
                max_class[cluster_id] = max(cluster.values()) #for
later determining purity
            #end for

            #purity -> if we have perfect clusters, the sum of the
max should equal 1000 (# of words)
            purity = sum(max_class.values())/1000.0*100
            print "-" * 80
            print "purity = {0:0.2f}%".format(purity)
            print "-" * 80
            break

        # write new centroids to file
        with open(CENTROIDS, 'w') as f:
            for centroid in centroid_points:
                f.writelines(','.join(map(str, centroid)) + '\n')
        f.close()
        i += 1
```

4.5 A

K=4 uniform random centroid-distributions over the 1000 words

In [15]:
```
import os
import numpy as np

K = 4
CENTROID = 'Centroids.txt'

try:
    os.remove(CENTROID)
except OSError:
    pass

centroid_points = []
for k in xrange(K):
    centroid_points.append(np.random.uniform(0.0, 0.40991, 1000)/10
00) #generate random centroid-distributions

with open(CENTROID, 'w+') as f:
        f.writelines(','.join(str(j) for j in i) + '\n' for i in ce
ntroid_points)

kmeans(K, centroid_points, "Output_45a")
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working.
working..
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working...

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working....
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working.....

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
```

```
working......
cluster distribution
------------------------------------------------------------------
--------------
iteration # 6
------------------------------------------------------------------
--------------
    k |   752 Human (%) |   91 Cyborg (%) |    54 Robot (%) | 103
Spammer (%)
------------------------------------------------------------------
--------------
    0 |   705 ( 93.75%) |     3 (  3.30%) |     2 (  3.70%) |    3
4 ( 33.01%)
    1 |    46 (  6.12%) |     0 (  0.00%) |    12 ( 22.22%) |    6
5 ( 63.11%)
    2 |     0 (  0.00%) |     0 (  0.00%) |     2 (  3.70%) |
0 (  0.00%)
    3 |     1 (  0.13%) |    88 ( 96.70%) |    38 ( 70.37%) |
4 (  3.88%)
------------------------------------------------------------------
--------------
purity = 86.00%
------------------------------------------------------------------
--------------
```

K=4 uniform random centroid distribution setting resulted in a purity score of 86.00%. Human (93.75% in cluster 0) and Cyborg (96.70% in cluster 0) classifications had the highest percentage of classifications in a cluster, while there was more uncertainty for the Robot (70.37% in cluster 3) and Spammer (63.11% in cluster 1) classifications. The groupings seem to indicate that Cyborg and Robot classifications overlap/share many similarities, and are grouped together in cluster 3.

4.5 B

K=2 perturbation-centroids, randomly perturbed from the aggregated (user-wide) distribution

In [16]:
```
import os
import numpy as np

K = 2
CENTROID = 'Centroids.txt'
output_file = "Output_45b.txt"

try:
    os.remove(output_file)
except OSError:
    pass

centroid_points = []
with open('topUsers_Apr-Jul_2014_1000-words_summaries.txt', 'r') as
f:
    for line in f:
        if line.startswith('ALL_CODES'):
            elements = line.strip().split(",")
            total = int(elements[2])
            array = [int(e) * 1.0 / total for e in elements[3:]]
            for k in xrange(K):
                centroid_points.append(array)
            break;

# Add Random Noise
centroid_points = centroid_points + np.random.sample(K * 1000).resh
ape(K, 1000)
for k in xrange(K):
    # Normalize Again
    centroid_points[k] = centroid_points[k] * 1.0 / np.sum(centroid
_points[k])

with open(CENTROID, 'w+') as f:
        f.writelines(','.join(str(j) for j in i) + '\n' for i in ce
ntroid_points)

kmeans(K, centroid_points, output_file)
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working.
working..
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working...

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
```

```
working....
cluster distribution
------------------------------------------------------------------
--------------
iteration # 4
------------------------------------------------------------------
--------------
    k |   752 Human (%) |   91 Cyborg (%) |    54 Robot (%) | 103
Spammer (%)
------------------------------------------------------------------
--------------
    0 |   751 ( 99.87%) |     3 (  3.30%) |    14 ( 25.93%) |    9
9 ( 96.12%)
    1 |     1 (  0.13%) |    88 ( 96.70%) |    40 ( 74.07%) |
4 (  3.88%)
------------------------------------------------------------------
--------------
purity = 83.90%
------------------------------------------------------------------
--------------
```

Using 2 perturbation-centroids instead of 4 ends up grouping Cyborg and Robot classifications together again, this time at a higher % for each. Spammer and Humans are grouped together with near 100% class membership for both in cluster 0. The purity score dropped only slightly, indicating that using 2 perturbation-centroid clusters accurately captured almost as much of the majority classes for each cluster as the 4 uniform random centroid-distribution setting. The K-Means algorithm also reached conversion at 4 iterations compared to 6 for 4.5A, indicating that the initialization of centroids using 2 perturbation centroids placed the centroids at a initial position closer to their eventual local minima position.

4.5 C

K=4 perturbation-centroids, randomly perturbed from the aggregated (user-wide) distribution

In [17]:
```
import os
import numpy as np

K = 4
CENTROID = 'Centroids.txt'
output_file = "Output_45c.txt"

try:
    os.remove(output_file)
except OSError:
    pass

centroid_points = []
with open('topUsers_Apr-Jul_2014_1000-words_summaries.txt', 'r') as
f:
    for line in f:
        if line.startswith('ALL_CODES'):
            elements = line.strip().split(",")
            total = int(elements[2])
            array = [int(e) * 1.0 / total for e in elements[3:]]
            for k in xrange(K):
                centroid_points.append(array)
            break;

# Add Random Noise
centroid_points = centroid_points + np.random.sample(K * 1000).resh
ape(K, 1000)
for k in xrange(K):
    # Normalize Again
    centroid_points[k] = centroid_points[k] * 1.0 / np.sum(centroid
_points[k])

with open(CENTROID, 'w+') as f:
        f.writelines(','.join(str(j) for j in i) + '\n' for i in ce
ntroid_points)

kmeans(K, centroid_points, output_file)
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
```

```
working.
working..
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working...

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working....
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working.....

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
```

```
working......
cluster distribution
-------------------------------------------------------------------
--------------
iteration # 6
-------------------------------------------------------------------
--------------
    k |   752 Human (%) |   91 Cyborg (%) |    54 Robot (%) | 103
Spammer (%)
-------------------------------------------------------------------
--------------
    0 |     0 (  0.00%) |     2 (  2.20%) |     9 ( 16.67%) |
0 (  0.00%)
    1 |   751 ( 99.87%) |     3 (  3.30%) |     9 ( 16.67%) |    9
9 ( 96.12%)
    2 |     0 (  0.00%) |    51 ( 56.04%) |     0 (  0.00%) |
0 (  0.00%)
    3 |     1 (  0.13%) |    35 ( 38.46%) |    36 ( 66.67%) |
4 (  3.88%)
-------------------------------------------------------------------
--------------
purity = 84.70%
-------------------------------------------------------------------
--------------
```

4 perturbation-centroids generated a model that resulted in classification groupings similar to 4.5A, except with Cyborg and Robot grouped in cluster 2 instead of cluster 0. This model also resulted in the highest purity score for the Spammer classification, but also the highest entropy for Cyborg and Robot classifications. This setting also took 6 iterations, which could be a function of the 2 additional centroids compared to 4.5B and the increased uncertainty of classifying the Cyborg and Robot points.

4.5 D

K=4 "trained" centroids, determined by the sums across the classes

In [18]:
```
import os
import numpy as np

K = 4 #number of centroids
CENTROID = 'Centroids.txt'
output_file = "Output_45d.txt" #create output file

try:
    os.remove(output_file)
except OSError:
    pass

centroid_points = []
with open('topUsers_Apr-Jul_2014_1000-words_summaries.txt', 'r') as
f:
    for line in f:
        if line.startswith('CODE'):
            elements = line.strip().split(",")
            total = int(elements[2]) #total
            array = [int(e) * 1.0 / total for e in elements[3:]] #s
um across classes
            centroid_points.append(array)

with open(CENTROID, 'w+') as f:
        f.writelines(','.join(str(j) for j in i) + '\n' for i in ce
ntroid_points)

kmeans(K, centroid_points, output_file)
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working.
working..
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working...

WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working....
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protoc
ols will be strict by default. It's recommended you run your job w
ith --strict-protocols or set up mrjob.conf as described at http
s://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protoco
ls
WARNING:mrjob.runner:
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.
WARNING:mrjob.step:MRJobStep has been renamed to MRStep. The old n
ame will be removed in v0.5.0.

working.....
cluster distribution
------------------------------------------------------------------
--------------
iteration # 5
------------------------------------------------------------------
--------------
    k |   752 Human (%) |   91 Cyborg (%) |    54 Robot (%) | 103
Spammer (%)
------------------------------------------------------------------
--------------
    0 |   749 ( 99.60%) |     3 (  3.30%) |    14 ( 25.93%) |    3
8 ( 36.89%)
    1 |     0 (  0.00%) |    51 ( 56.04%) |     0 (  0.00%) |
0 (  0.00%)
    2 |     1 (  0.13%) |    37 ( 40.66%) |    40 ( 74.07%) |
4 (  3.88%)
    3 |     2 (  0.27%) |     0 (  0.00%) |     0 (  0.00%) |    6
1 ( 59.22%)
------------------------------------------------------------------
--------------
purity = 90.10%
------------------------------------------------------------------
--------------
```

4 trained centroids results in a 90.10% purity score, which is marginally better than the 4 perturbation centroid setting. Human classification is almost perfect, which generates the higher overall purity score due to there being many more true Human classifications and having an almost perfect purity score for Human. Only 5 iterations to reach the threshold convergence even with 4 centroids, which makes sense because "trained" centroids should lead to initialization at points closer to local minimas.