MIDS-W261-2016-HW-Week04-Lane

June 11, 2016

1 DATASCI W261: Machine Learning at Scale

1.1 Assignment Week 4

Jackson Lane (jelane@berkeley.edu) W261-3 # === INSTRUCTIONS for SUBMISSIONS === Follow the instructions for submissions carefully.

Submit your homework via the following form by 8:00AM of the following Tuesday (West Coast Time):
https://docs.google.com/forms/d/1ZOr9RnIe_A06AcZDB6K1mJN4vrLeSmS2PD6Xm3eOiis/viewform?usp=send_form

2 === Week 4 ASSIGNMENTS ===

3 HW 4.0.

What is MrJob? How is it different to Hadoop MapReduce? MRJob is an abstraction that gives an abstraction of the MapReduce programming framework. It was developed by Yelp but is now available for free to the public.

The main difference is that MRJob is an abstraction while Hadoop MapReduce is an actual implementation of the MapReduce programming paradigm. You can run MRJob against multiple different MapReduce backends (Amazon EMR, Hadoop MapReduce, and Google Cloud Dataproc), where as jobs written in Hadoop MapReduce can only run on Hadoop MapReduce. Although MRJob does come with its own simple MapReduce implementation, the MRJob production use case is not to run jobs locally against this backend, but rather to run against a separate MapReduce implementation backend.

MrJob also provides a simpler way to write MapReduce jobs in Python versus someting like Hadoop Streaming. Whereas in Hadoop Streaming you would need to write each mapper, combiner, and reducer as its own class file with a separate invocation for each MR step, in MRJob you can write everything in a single Python class file. MRJob also takes care of splitting up the key and value fields in the MapReduce outputs, whereas in Hadoop Streaming you would need to split up the key and value fields manually.

The drawback is that MRJob does not have as robust capabilities as Hadoop MapReduce. In general, abstractions do not cover every funtionality of their backend.

What are the mapper_init, mapper_final(), combiner_final(), reducer_final() methods? When are they called? The mapper_init (and the reduce_init) functions are all functions called once per node at the begining of the Map (and Reduce) stage in the MR Job. They are used to initialize counter variables or load other data sets for use in the main mapper method.

The mapper_final, combiner_final, and reducer_final methods are called after the mapper, combiner, and reducer phases, respectively. They are used to tear down variables and clean up.

4 HW 4.1

What is serialization in the context of MrJob or Hadoop? Serialization is how MRJob converts data to and from byes between each of the phases in a MapReduce job.

When it used in these frameworks? It's used to convert the input text into key value pairs for the mappers, to write mapper output to disk, to convert bytes received from the shuffle toto keys and values for reducer input, and to write key value pairs back to text for output.

What is the default serialization mode for input and outputs for MrJob? The default serialization mode for input is RawValueProtocol. The default serialization mode for both internal communication and outputs is JSONProtocol.

5 HW 4.2: 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 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 [446]: #!/usr/bin/python
          import sys,re

          processed = open('Processed-anonymous-msweb.data', 'w')
          urls = open('urls.data', 'w')

          customer= "00000"
          with open ("anonymous-msweb.data", "r") as myfile:
              for line in myfile.readlines():
                  line = line.strip()
                  #Keep track of the most recent customer ID
                  if(line[0]=='C'):
                      customer = line[line.rfind(","):]
                      continue
                  #Output visit information joined with customer ID
                  if(line[0]=='V'):
                      processed.write(line+',C'+customer+'\n')
                      continue
                  if(line[0]=='A'):
                      urls.write(line+"\n")
```

```
processed.close()
urls.close()
```

6 HW 4.3: Find the 5 most frequently visited pages using MrJob from the output of 4.2 (i.e., transfromed log file).

In [447]:
```
%%writefile MRJob4_3.py
#!/usr/bin/python
# MRJob4_3.py
# Author: Jackson Lane
# Description: MRJob code to find the 5 most frequently visited pages from the output of 4.2
# Has two MR steps:
# The first step aggregates the number of visits to each site
# The second step get sthe top 5 sites with the most visits

from mrjob.job import MRJob, MRStep
import csv
import sys


class MRJob4_3(MRJob):

    def mapper_visitcount(self, _, line):
        # Passes on the page id and the visit count to the reducer
        line = line.strip().split(",")
        yield line[1], int(line[2])

    def reducer_visitcount(self, pageID, counts):
        # sums up the visit counts for each page
        yield pageID, sum(counts)


    def reducer_top5_init(self):
        # We want to maintain a counter to only print out the top 5 sites
        self.currentrank = 5

    def reducer_top5(self, pageID, visit_counts):
        # Decrement counter and yield current site and visits
        if self.currentrank > 0:
            self.currentrank -= 1
            yield pageID, sum(visit_counts)

    def steps(self):
        return [MRStep(mapper=self.mapper_visitcount,reducer=self.reducer_visitcount),
                MRStep(reducer_init=self.reducer_top5_init,reducer=self.reducer_top5,
                jobconf={
                    "stream.num.map.output.key.fields":"2",
                    "mapreduce.job.output.key.comparator.class":
                        "org.apache.hadoop.mapred.lib.KeyFieldBasedComparator",
                    "mapreduce.partition.keycomparator.options":"-k2,2nr"
                          })]
```

```
        if __name__ == '__main__':
            MRJob4_3.run()
```

```
Overwriting MRJob4_3.py
```

```
In [448]: %reload_ext autoreload
          %autoreload 2
          from MRJob4_3 import MRJob4_3

          mr_job = MRJob4_3(args=['Processed-anonymous-msweb.data','-r', 'hadoop', '--strict-protocols']
          with mr_job.make_runner() as runner:
              runner.run()
              print ("PageID\tCount")

              for line in runner.stream_output():
                  line = mr_job.parse_output_line(line)
                  line=[str(i) for i in line]
                  print '\t'.join(line)
```

```
PageID      Count
1008        10836
1034        9383
1004        8463
1018        5330
1017        5108
```

7 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 [449]: %%writefile MRJob4_4.py
          #!/usr/bin/python
          # MRJob4_4.py
          # Author: Jackson Lane
          # Description: MRJob code to find each site's most frequent visitor.
          # Has two MR steps:
          # The first step aggregates the number of visits to each site by each customer
          # The second step gets the top visitor for each site
          # I've modified the visit count data to


          from mrjob.job import MRJob, MRStep
          import mrjob
          import csv
          import sys


          class MRJob4_4(MRJob):

              def mapper_visitcount(self, _, line):
                  # Passes on the page id and the visit count to the reducer
```

```
            line = line.strip().split(",")
            yield line[1]+","+line[4],line[2]

        def reducer_visitcount(self, keys, counts):
            # sums up the visit counts for each page
            pageID,visitorID = keys.split(",")
            yield pageID, str(visitorID) +"\t"+str(sum([int(i) for i in counts]))

        # The reducer init here creates a dictionary with each site's URL
        # It gets this data from the "urls.data" file sent to each reducer in the MRJob config
        def reducer_top_init(self):
            with open("urls.data") as urls:
                self.urls = {}
                for line in urls.readlines():
                    fields = line.split(",")
                    self.urls[fields[1]] = "www.microsoft.com" + fields[4].strip().strip('"')

        def reducer_top(self, pageID, visitor):
                yield self.urls[pageID]+"\t"+pageID, visitor.next()

        def steps(self):
            return [MRStep(mapper=self.mapper_visitcount,reducer=self.reducer_visitcount),
                    MRStep(reducer_init=self.reducer_top_init,reducer=self.reducer_top,
                    jobconf={
                        "stream.num.map.output.key.fields":"3",
                        "mapreduce.job.output.key.comparator.class":
                            "org.apache.hadoop.mapred.lib.KeyFieldBasedComparator",
                        "mapreduce.partition.keycomparator.options":"-k3,3nr"
                              })]


    if __name__ == '__main__':
        MRJob4_4.run()

Overwriting MRJob4_4.py
```

Driver Class

```
In [452]: %reload_ext autoreload
          %autoreload 2

          from MRJob4_4 import MRJob4_4

          mr_job = MRJob4_4(args=['Processed-anonymous-msweb.data','r','hadoop', '--file',"urls.data",'
          with mr_job.make_runner() as runner:
              runner.run()
              print 'URL\t\t\t\tPageID\tVisitorID\tVisits'
              for line in runner.stream_output():
                  line =  mr_job.parse_output_line(line)
                  print '\t'.join(line)

URL                                PageID        VisitorID        Visits
www.microsoft.com/sitebuilder        1026        10016        1
www.microsoft.com/intdev        1027        10017        1
www.microsoft.com/oledev        1028        10017        1
```

```
www.microsoft.com/clipgallerylive        1029        10019        1
www.microsoft.com/ntserver        1030        10019        1
www.microsoft.com/msoffice        1031        10019        1
www.microsoft.com/games        1032        10019        1
www.microsoft.com/logostore        1033        10019        1
www.microsoft.com/ie        1034        10020        1
www.microsoft.com/windowssupport        1035        10021        1
www.microsoft.com/organizations        1036        10021        1
www.microsoft.com/windows95        1037        10021        1
www.microsoft.com/sbnmember        1038        10021        1
www.microsoft.com/isp        1039        10021        1
www.microsoft.com/office        1040        10021        1
www.microsoft.com/workshop        1041        10021        1
www.microsoft.com/vstudio        1042        10021        1
www.microsoft.com/smallbiz        1043        10021        1
www.microsoft.com/mediadev        1044        10024        1
www.microsoft.com/netmeeting        1045        10025        1
www.microsoft.com/iesupport        1046        10027        1
www.microsoft.com/publisher        1048        10030        1
www.microsoft.com/supportnet        1049        10031        1
www.microsoft.com/macoffice        1050        10032        1
www.microsoft.com/scheduleplus        1051        10035        1
www.microsoft.com/word        1052        10035        1
www.microsoft.com/visualj        1053        10035        1
www.microsoft.com/exchange        1054        10036        1
www.microsoft.com/kids        1055        10037        1
www.microsoft.com/sports        1056        10037        1
www.microsoft.com/powerpoint        1057        10038        1
www.microsoft.com/referral        1058        10039        1
www.microsoft.com/sverige        1059        10047        1
www.microsoft.com/msword        1060        10053        1
www.microsoft.com/promo        1061        10058        1
www.microsoft.com/msaccess        1062        10065        1
www.microsoft.com/intranet        1063        10067        1
www.microsoft.com/activeplatform        1064        10068        1
www.microsoft.com/java        1065        10068        1
www.microsoft.com/musicproducer        1066        10068        1
www.microsoft.com/frontpage        1067        10068        1
www.microsoft.com/vbscript        1068        10068        1
www.microsoft.com/windowsce        1069        10068        1
www.microsoft.com/activex        1070        10068        1
www.microsoft.com/automap        1071        10068        1
www.microsoft.com/vinterdev        1072        10068        1
www.microsoft.com/taiwan        1073        10078        1
www.microsoft.com/ntworkstation        1074        10085        1
www.microsoft.com/jobs        1075        10104        1
www.microsoft.com/ntwkssupport        1076        10107        1
www.microsoft.com/msofficesupport        1077        10109        1
www.microsoft.com/ntserversupport        1078        10122        1
www.microsoft.com/australia        1079        10122        1
www.microsoft.com/brasil        1080        10126        1
www.microsoft.com/accessdev        1081        10127        1
www.microsoft.com/access        1082        10127        1
www.microsoft.com/msaccesssupport        1083        10127        1
```

```
www.microsoft.com/uk        1084        10132        1
www.microsoft.com/exchangesupport        1085        10132        1
www.microsoft.com/oem        1086        10132        1
www.microsoft.com/proxy        1087        10132        1
www.microsoft.com/outlook        1088        10132        1
www.microsoft.com/officereference        1089        10132        1
www.microsoft.com/gamessupport        1090        10133        1
www.microsoft.com/hwdev        1091        10142        1
www.microsoft.com/vfoxpro        1092        10150        1
www.microsoft.com/vba        1093        10156        1
www.microsoft.com/mshome        1094        10156        1
www.microsoft.com/catalog        1095        10156        1
www.microsoft.com/mspress        1096        10156        1
www.microsoft.com/latam        1097        10156        1
www.microsoft.com/devonly        1098        10157        1
www.microsoft.com/cio        1099        10159        1
www.microsoft.com/education        1100        10165        1
www.microsoft.com/oledb        1101        10166        1
www.microsoft.com/homeessentials        1102        10168        1
www.microsoft.com/works        1103        10168        1
www.microsoft.com/hk        1104        10191        1
www.microsoft.com/france        1105        10197        1
www.microsoft.com/cze        1106        10198        1
www.microsoft.com/slovakia        1107        10198        1
www.microsoft.com/teammanager        1108        10205        1
www.microsoft.com/technet        1109        10205        1
www.microsoft.com/mastering        1110        10208        1
www.microsoft.com/ssafe        1111        10208        1
www.microsoft.com/canada        1112        10208        1
www.microsoft.com/security        1113        10215        1
www.microsoft.com/servad        1114        10216        1
www.microsoft.com/hun        1115        10216        1
www.microsoft.com/switzerland        1116        10225        1
www.microsoft.com/sidewinder        1117        10228        1
www.microsoft.com/sql        1118        10235        1
www.microsoft.com/corpinfo        1119        10240        1
www.microsoft.com/switch        1120        10241        1
www.microsoft.com/magazine        1121        10241        1
www.microsoft.com/mindshare        1122        10243        1
www.microsoft.com/germany        1123        10254        1
www.microsoft.com/industry        1124        10263        1
www.microsoft.com/imagecomposer        1125        10269        1
www.microsoft.com/mediamanager        1126        10272        1
www.microsoft.com/netshow        1127        10286        1
www.microsoft.com/msf        1128        10286        1
www.microsoft.com/ado        1129        10290        1
www.microsoft.com/syspro        1130        10306        1
www.microsoft.com/moneyzone        1131        10316        1
www.microsoft.com/msmoneysupport        1132        10316        1
www.microsoft.com/frontpagesupport        1133        10319        1
www.microsoft.com/backoffice        1134        10335        1
www.microsoft.com/mswordsupport        1135        10339        1
www.microsoft.com/usa        1136        10348        1
www.microsoft.com/mscorp        1137        10348        1
```

```
www.microsoft.com/mind        1138        10351        1
www.microsoft.com/k-12        1139        10362        1
www.microsoft.com/netherlands        1140        10363        1
www.microsoft.com/europe        1141        10372        1
www.microsoft.com/southafrica        1142        10372        1
www.microsoft.com/workshoop        1143        10381        1
www.microsoft.com/devnews        1144        10406        1
www.microsoft.com/vfoxprosupport        1145        10418        1
www.microsoft.com/msp        1146        10429        1
www.microsoft.com/msft        1147        10438        1
www.microsoft.com/channel_resources        1148        10468        1
www.microsoft.com/adc        1149        10471        1
www.microsoft.com/infoserv        1150        10473        1
www.microsoft.com/mspowerpointsupport        1151        10482        1
www.microsoft.com/rus        1152        10486        1
www.microsoft.com/venezuela        1153        10500        1
www.microsoft.com/project        1154        10564        1
www.microsoft.com/sidewalk        1155        10606        1
www.microsoft.com/powered        1156        10624        1
www.microsoft.com/win32dev        1157        10627        1
www.microsoft.com/imedia        1158        10632        1
www.microsoft.com/transaction        1159        10661        1
www.microsoft.com/visualcsupport        1160        10669        1
www.microsoft.com/workssupport        1161        10677        1
www.microsoft.com/infoservsupport        1162        10699        1
www.microsoft.com/opentype        1163        10744        1
www.microsoft.com/smsmgmt        1164        10752        1
www.microsoft.com/poland        1165        10784        1
www.microsoft.com/mexico        1166        10788        1
www.microsoft.com/hwtest        1167        10791        1
www.microsoft.com/salesinfo        1168        10797        1
www.microsoft.com/msproject        1169        10797        1
www.microsoft.com/mail        1170        10821        1
www.microsoft.com/merchant        1171        10828        1
www.microsoft.com/belgium        1172        10834        1
www.microsoft.com/moli        1173        10842        1
www.microsoft.com/nz        1174        10866        1
www.microsoft.com/msprojectsupport        1175        10888        1
www.microsoft.com/jscript        1176        10932        1
www.microsoft.com/events        1177        10951        1
www.microsoft.com/msdownload.        1178        11008        1
www.microsoft.com/colombia        1179        11027        1
www.microsoft.com/slovenija        1180        11035        1
www.microsoft.com/kidssupport        1181        11044        1
www.microsoft.com/fortran        1182        11090        1
www.microsoft.com/italy        1183        11111        1
www.microsoft.com/msexcelsupport        1184        11134        1
www.microsoft.com/sna        1185        11142        1
www.microsoft.com/college        1186        11150        1
www.microsoft.com/odbc        1187        11173        1
www.microsoft.com/korea        1188        11190        1
www.microsoft.com/internet        1189        11243        1
www.microsoft.com/repository        1190        11287        1
www.microsoft.com/management        1191        11331        1
```

```
www.microsoft.com/visualjsupport        1192        11359        1
www.microsoft.com/offdevsupport        1193        11367        1
www.microsoft.com/china        1194        11372        1
www.microsoft.com/portugal        1195        11429        1
www.microsoft.com/ie40        1196        11431        1
www.microsoft.com/sqlsupport        1197        11444        1
www.microsoft.com/pictureit        1198        11482        1
www.microsoft.com/feedback        1199        11644        1
www.microsoft.com/benelux        1200        11674        1
www.microsoft.com/hardware        1201        11800        1
www.microsoft.com/advtech        1202        11802        1
www.microsoft.com/danmark        1203        11806        1
www.microsoft.com/msscheduleplus        1204        11904        1
www.microsoft.com/hardwaresupport        1205        11917        1
www.microsoft.com/select        1206        12011        1
www.microsoft.com/icp        1207        12135        1
www.microsoft.com/israel        1208        12177        1
www.microsoft.com/turkey        1209        12239        1
www.microsoft.com/snasupport        1210        12359        1
www.microsoft.com/smsmgmtsupport        1211        12395        1
www.microsoft.com/worldwide        1212        12421        1
www.microsoft.com/corporate_solutions        1213        12472        1
www.microsoft.com/finserv        1214        12515        1
www.microsoft.com/developer        1215        12577        1
www.microsoft.com/vrml        1216        12666        1
www.microsoft.com/ireland        1217        12675        1
www.microsoft.com/publishersupport        1218        12714        1
www.microsoft.com/ads        1219        12746        1
www.microsoft.com/macofficesupport        1220        12795        1
www.microsoft.com/mstv        1221        12815        1
www.microsoft.com/msofc        1222        12819        1
www.microsoft.com/finland        1223        12828        1
www.microsoft.com/atec        1224        13041        1
www.microsoft.com/piracy        1225        13061        1
www.microsoft.com/msschedplussupport        1226        13179        1
www.microsoft.com/argentina        1227        13235        1
www.microsoft.com/vtest        1228        13266        1
www.microsoft.com/uruguay        1229        13510        1
www.microsoft.com/mailsupport        1230        13541        1
www.microsoft.com/win32devsupport        1231        13918        1
www.microsoft.com/standards        1232        13926        1
www.microsoft.com/vbscripts        1233        14363        1
www.microsoft.com/off97cat        1234        14418        1
www.microsoft.com/onlineeval        1235        14522        1
www.microsoft.com/globaldev        1236        14738        1
www.microsoft.com/devdays        1237        14764        1
www.microsoft.com/exceldev        1238        15247        1
www.microsoft.com/msconsult        1239        15361        1
www.microsoft.com/thailand        1240        15461        1
www.microsoft.com/india        1241        15820        1
www.microsoft.com/msgarden        1242        16289        1
www.microsoft.com/usability        1243        16904        1
www.microsoft.com/devwire        1244        16967        1
www.microsoft.com/ofc        1245        16999        1
```

```
www.microsoft.com/gamesdev        1246        17120        1
www.microsoft.com/wineguide        1247        18240        1
www.microsoft.com/softimage        1248        18347        1
www.microsoft.com/fortransupport        1249        18384        1
www.microsoft.com/middleeast        1250        18534        1
www.microsoft.com/referencesupport        1251        18941        1
www.microsoft.com/giving        1252        19483        1
www.microsoft.com/worddev        1253        19746        1
www.microsoft.com/ie3        1254        20190        1
www.microsoft.com/msmq        1255        20277        1
www.microsoft.com/sia        1256        20832        1
www.microsoft.com/devvideos        1257        21184        1
www.microsoft.com/peru        1258        21399        1
www.microsoft.com/controls        1259        21424        1
www.microsoft.com/trial        1260        21894        1
www.microsoft.com/diyguide        1261        22485        1
www.microsoft.com/chile        1262        24951        1
www.microsoft.com/services        1263        26122        1
www.microsoft.com/se_partners        1264        26801        1
www.microsoft.com/ssafesupport        1265        26811        1
www.microsoft.com/licenses        1266        26815        1
www.microsoft.com/caribbean        1267        27482        1
www.microsoft.com/javascript        1268        27503        1
www.microsoft.com/business        1269        28044        1
www.microsoft.com/developr        1270        28493        1
www.microsoft.com/mdsn        1271        28493        1
www.microsoft.com/softlib        1272        28493        1
www.microsoft.com/mdn        1273        28493        1
www.microsoft.com/pdc        1274        28493        1
www.microsoft.com/security.        1275        28903        1
www.microsoft.com/vtestsupport        1276        29654        1
www.microsoft.com/stream        1277        30111        1
www.microsoft.com/hed        1278        30460        1
www.microsoft.com/msgolf        1279        31062        1
www.microsoft.com/music        1280        33424        1
www.microsoft.com/intellimouse        1281        37099        1
www.microsoft.com/home        1282        39877        1
www.microsoft.com/cinemania        1283        41033        1
www.microsoft.com/partner        1284        41108        1
www.microsoft.com/train_cert        1295        10028        1
www.microsoft.com/regwiz        1000        10001        1
www.microsoft.com/support        1001        10001        1
www.microsoft.com/athome        1002        10001        1
www.microsoft.com/kb        1003        10002        1
www.microsoft.com/search        1004        10003        1
www.microsoft.com/norge        1005        10004        1
www.microsoft.com/misc        1006        10005        1
www.microsoft.com/ie_intl        1007        10007        1
www.microsoft.com/msdownload        1008        10009        1
www.microsoft.com/windows        1009        10009        1
www.microsoft.com/vbasic        1010        10010        1
www.microsoft.com/officedev        1011        10010        1
www.microsoft.com/outlookdev        1012        10010        1
www.microsoft.com/vbasicsupport        1013        10010        1
```

```
www.microsoft.com/officefreestuff        1014        10010        1
www.microsoft.com/msexcel        1015        10011        1
www.microsoft.com/excel        1016        10011        1
www.microsoft.com/products        1017        10011        1
www.microsoft.com/isapi        1018        10011        1
www.microsoft.com/mspowerpoint        1019        10011        1
www.microsoft.com/msdn        1020        10012        1
www.microsoft.com/visualc        1021        10012        1
www.microsoft.com/truetype        1022        10013        1
www.microsoft.com/spain        1023        10014        1
www.microsoft.com/iis        1024        10015        1
www.microsoft.com/gallery        1025        10016        1
```

8 HW 4.5 Clustering Tweet Dataset

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. . .)

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 (generate 1000 random numbers and normalize the vectors)
(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. Use use the (row-normalized) class-level aggregates as 'trained' starting centroids (i.e., the training is already done for you!). Note that you do not have to compute the aggregated distribution or the class-aggregated distributions, which are rows in the auxiliary file:

For experiments A, B, C and D 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.

In [440]:
```
%%writefile Kmeans.py
# Kmeans code takan from Week 5 Readings
# Modified to accommodate 1000 fields instead of 4
# And to take variable k
# And to drop first 3 fields from input lines

from numpy import argmin, array, random
from mrjob.job import MRJob, MRStep
from itertools import chain
```

```
#Calculate find the nearest centroid for data point
def MinDist(datapoint, centroid_points):
    datapoint = array(datapoint)
    centroid_points = array(centroid_points)
    diff = datapoint - centroid_points
    diffsq = diff*diff
    # Get the nearest centroid for each instance
    minidx = argmin(list(diffsq.sum(axis = 1)))
    return minidx

#Check whether centroids converge
def stop_criterion(centroid_points_old, centroid_points_new,T):
    oldvalue = list(chain(*centroid_points_old))
    newvalue = list(chain(*centroid_points_new))
    Diff = [abs(x-y) for x, y in zip(oldvalue, newvalue)]
    Flag = True
    for i in Diff:
        if(i>T):
            Flag = False
            break
    return Flag

class MRKmeans(MRJob):
    centroid_points=[]
    k=4
    def steps(self):
        return [
            MRStep(mapper_init = self.mapper_init, mapper=self.mapper,combiner = self.combine
               ]
    def mapper_init(self):
        #load centroids info from file
        self.centroid_points = [map(float,s.split('\n')[0].split(',')) for s in open("Centroi
        self.k = len(self.centroid_points)
    #Mapper to assign each tweet to the closest centroid
    def mapper(self, _, line):
        line =line.split(",")
        D = map(float,line[3:])
        yield int(MinDist(D,self.centroid_points)), [D,1]
    #Combine sum of data points locally
    def combiner(self, idx, inputdata):
        sums = [0 for i in range(1000)]
        num = 0
        for D,n in inputdata:
            num = num + n
            sums = [sums[i] + int(D[i]) for i in range(1000)]
        yield idx,(sums,num)
    #Aggregate sum for each cluster and then calculate the new centroids
    def reducer(self, idx, inputdata):
        centroids = [[float(0)]*1000] * self.k
        num = [0]*self.k
        for D, n in inputdata:
            num[idx] = num[idx] + n
            centroids[idx] = [centroids[idx][i] + D[i] for i in range(1000)]
```

```
            for i in range(1000):
                centroids[idx][i] = centroids[idx][i]/num[idx]
            yield idx,centroids[idx]

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

```
Overwriting Kmeans.py
```

In [433]:
```
# Functions to generate the different types of centroid point

# The below was taken from the homework instructions.
# I changed the name of the function to "perturbed" for clarity

##############################################################################################
##Geneate random initial centroids around the global aggregate
##Part (B) and (C) of this question
##############################################################################################
def perturbed(k):
    counter = 0
    for line in open("topUsers_Apr-Jul_2014_1000-words_summaries.txt").readlines():
        if counter == 2:
            data = re.split(",",line)
            globalAggregate = [float(data[i+3])/float(data[2]) for i in range(1000)]
        counter += 1
    #perturb the global aggregate for the four initializations
    centroids = []
    for i in range(k):
        rndpoints = random.sample(1000)
        peturpoints = [rndpoints[n]/10+globalAggregate[n] for n in range(1000)]
        centroids.append(peturpoints)
        total = 0    # We need to find the data-wide centroid first
    # Add random noise to this data-wide centroid to get the second centroi
        for j in range(len(centroids[i])):
            total += centroids[i][j]
        for j in range(len(centroids[i])):
            centroids[i][j] = centroids[i][j]/total
    return centroids

# Function to generate uniform centroids.
# Pick k random tweets from the corpus to serve as our inital centroids
def uniform(k):
        data=[]
        with open('topUsers_Apr-Jul_2014_1000-words.txt', 'r') as myfile:
            for line in myfile.readlines():
                line = line.strip()
                line = line.split(",")
                # Collect the word count fields into an array
                data.append(line[3:])
        #We now randomly select k tweets to be our centroids
        return [map(int,data[random.randint(0,1000)]) for i in range(k)]

# Function to get the trained centroids from the 1000 word summaries file
def trained(k):
    with open('topUsers_Apr-Jul_2014_1000-words_summaries.txt','r') as myfile:
```

```
        centroids = []
        #Skip first two lines (column names and totals)
        myfile.readline()
        myfile.readline()

        #The remaining four lines in the summaries text file correspond to
        #the word counts for each of the four types of users.
        for line in myfile.readlines():
            line = line.strip()
            line = line.split(",")
            # Get the per tweet average for each of the word counts
            point = [int(i)/float(line[2]) for i in line[3:]]
            centroids.append(point)
        # Return only the first k of the centroids
        return centroids[:k]
```

In [434]:
```
#Functions to report the goodness of fit and purity of the generated k-means clusters

#Reuse the MinDist function from the KMeans MRJob
def MinDist(datapoint, centroid_points):
    datapoint = array(datapoint)
    centroid_points = array(centroid_points)
    diff = datapoint - centroid_points
    diffsq = diff*diff
    # Get the nearest centroid for each instance
    minidx = argmin(list(diffsq.sum(axis = 1)))
    return minidx


def reportPurity(centroids,k):
    # Confusion matrix and summary variables
    confuse_matrix = [[0 for x in range(k)] for y in range(4)]
    predicted_counts = [0 for x in range(k)]
    actual_counts = [0 for x in range(4)]
    # Assign each tweet to the closest cluster, check if cluster matches actual code
    with open('topUsers_Apr-Jul_2014_1000-words.txt') as myfile:
        for line in myfile.readlines():
            line = line.strip().split(",")
            # As before, ignore the first three fields and only consider the actual word coun
            tweet = [float(i) for i in line[3:]]
            # Get the closest cluster
            predicted = int(MinDist(tweet, centroids))
            # Get which code the tweet actually belongs to
            actual = int(line[1])
            # Update the confusion matrix and totals
            confuse_matrix[actual][predicted] += 1
            predicted_counts[predicted] += 1
            actual_counts[actual] += 1

    # With confusion matrix and prediction counts, calculate purity
    purity = sum([max([confuse_matrix[i][j] for i in range(4)])  for j in range(k)])/float(su


    # Print out the findings
```

```
    labels = ["Human","Cyborg","Robot","Spammer"]
    print "\tCluster ID"
    print "Actual\t|\t" +'\t'.join(map(str,range(k))) + "\t|Total"
    print "-----------------------------------------------------------------"
    for i in range(4):
        print labels[i] + '\t|\t'+ '\t'.join(map(str,confuse_matrix[i])) + "\t|" + str(actual_
    print "-----------------------------------------------------------------"
    print "Total\t|\t" + "\t".join(map(str,predicted_counts))

    print
    print "Purity Score:",purity
```

Driver class

```
In [443]: # Since we'll need to run this 4 times, I've turned the driver code from the Week 5 Kmeans co
          from numpy import random
          from Kmeans import MRKmeans, stop_criterion
          mr_job = MRKmeans(args=['topUsers_Apr-Jul_2014_1000-words.txt', '-r', 'hadoop','--file', 'Cen



          def run_kmeans(centroid_type, k):


              # First we have to generate the centroid points.
              centroid_points = eval(centroid_type+"("+str(k)+")")
              num = []

              with open('Centroids.txt', 'w+') as myfile:
                      myfile.writelines(','.join(str(j) for j in i) + '\n' for i in centroid_points)

              # Update centroids iteratively
              i = 0
              while(1):
                  # save previous centoids to check convergency
                  centroid_points_old = centroid_points[:]
                  with mr_job.make_runner() as runner:
                      runner.run()
                      # stream_output: get access of the output
                      for line in runner.stream_output():
                          key,value =  mr_job.parse_output_line(line)
                          centroid_points[key] = value
                  i = i + 1
                  if(stop_criterion(centroid_points_old,centroid_points,0.01)):
                      break
                  with open('Centroids.txt', 'w+') as myfile:
                      myfile.writelines(','.join(str(j) for j in i) + '\n' for i in centroid_points)
              print "Convergence after", i, "iterations"

              #Report the purity and summary
              reportPurity(centroid_points,k)
```

4.5 A

```
In [444]: run_kmeans("uniform", 4)
```

```
Convergence after 17 iterations
        Cluster ID
Actual         |        0        1        2        3        |Total
-------------------------------------------------------------------
Human          |        0        0        0        752      |752
Cyborg          |        26        12        4        49       |91
Robot          |        3        10        4        37       |54
Spammer          |        0        0        0        103       |103
-------------------------------------------------------------------
Total          |        29        22        8        941

Purity Score: 0.794
```

4.5 B

```
In [445]: run_kmeans("perturbed", 2)

Convergence after 11 iterations
        Cluster ID
Actual         |        0        1        |Total
-------------------------------------------------------------------
Human          |        752        0        |752
Cyborg          |        76        15        |91
Robot          |        45        9        |54
Spammer          |        103        0        |103
-------------------------------------------------------------------
Total          |        976        24

Purity Score: 0.767
```

4.5 C

```
In [438]: run_kmeans("perturbed", 4)

Convergence after 24 iterations
        Cluster ID
Actual         |        0        1        2        3        |Total
-------------------------------------------------------------------
Human          |        0        0        752        0        |752
Cyborg          |        4        0        50        37        |91
Robot          |        4        4        44        2        |54
Spammer          |        0        0        103        0        |103
-------------------------------------------------------------------
Total          |        8        4        949        39

Purity Score: 0.797
```

4.5 D

```
In [439]: run_kmeans("trained", 4)

Convergence after 19 iterations
        Cluster ID
Actual         |        0        1        2        3        |Total
-------------------------------------------------------------------
```

```
Human         |        752        0        0        0        |752
Cyborg         |        31        13        0        47        |91
Robot         |        40        3        4        7        |54
Spammer         |        103        0        0        0        |103
-----------------------------------------------------------------
Total         |        926        16        4        54

Purity Score: 0.816
```

Discussion The trained centroids has the best Purity Score, which makes sense as the trained centroids were derived formulaically from the "training" data itself, the corpus of tweets. But this means that the scores from trained centroids will probably not be generalizable if we apply these clusters to a different cluster, as it's likely the trained centroids are an overfitting of the training data.

The other three types of centroids were all generated with some randomness and thus have lower purity scores than the trained centroids, but their scores will likely be more generalizable than those of the trained centroids to different corpuses. The centroids generated with k=2 in particular has a lower purity because it's trying to put 4 different types of data into just two clusters, so 100% purity is mathematically impossible.