Data Science W261: Machine Learning at Scale

Safyre Anderson

safyre@berkeley.edu

January 27, 2016 8am

W261-3

Week 2 HW

HW2.0.

What is a race condition in the context of parallel computation? Give an example.

What is MapReduce?

How does it differ from Hadoop?

Which programming paradigm is Hadoop based on? Explain and give a simple example in code and show the code running.

Race Condition:

A race condition refers to the issue of timing the completion and sequence of parallel jobs. For example, let's say there is one round of parrallel jobs is running and there is a subsequent round of parallel jobs. If not all the jobs from the first round of parallel jobs complete and the second round begins prematurely, the job will not complete correctly. This is known as a race condition. Hadoop HDFS was designed to abstract the prevention of race conditions and communication between parallel processes from developers.

MapReduce

MapReduce in the broadest sense is a functional programming-inpired parallel processing framework. It consists of at least two major steps: map and reduce (though subsequent MapReduce jobs can follow).

Hadoop

Hadoop is a Java-based framework for distributed data storage and processing that implements its own MapReduce framework. However, Hadoop is also a distributed file system that consists of a master 'namenode' as well as at least one 'datanode'. The namenode keeps track of where data are stored as well as the implementation on MapReduce Jobs. The datanode(s) store typically 3 copies of the data in chunks which are scattered between the datanodes for redundancy and disaster recovery.

Programming Paradigm

In [23]:
```
!source .bashrc
!echo $HADOOP_HOME
```

```
/bin/sh: 1: source: not found
/usr/local/hadoop
```

HW2.1. Sort in Hadoop MapReduce

Given as input: Records of the form <integer, "NA"\>, where integer is any integer, and "NA" is just the empty string. Output: sorted key value pairs of the form in decreasing order; what happens if you have multiple reducers? Do you need additional steps? Explain.

Write code to generate N random records of the form <integer, "NA">. Let N = 10,000. Write the python Hadoop streaming map-reduce job to perform this sort. Display the top 10 biggest numbers. Display the 10 smallest numbers

I used 1 node Hadoop cluster AWS instance:

```
alias aws_hadoop_master="ssh -i "~/.ssh/***.pem" ec2-user@ec2-54-213-63-
253.us-west-2.compute.amazonaws.com"
```

In [14]:
```
!pwd
```

```
/home/ubuntu
```

In [9]:
```
%%writefile rand_num.py
#!/usr/bin/python

import random
random.seed(0)
count = 0
N = 10000
while count < N:
    print str(random.randint(0,N)) +"\t" + "NA"
    count +=1
```

```
Overwriting rand_num.py
```

In [10]:
```
!rm random_numbers.txt
!python rand_num.py > random_numbers.txt
```

As a sanity check, I wanted to print out the top 10 rows and make sure there were only 10000 numbers (one on each line) generated.

In [11]:
```
!head random_numbers.txt
!wc -l random_numbers.txt
```

```
8445    NA
7580    NA
4206    NA
2589    NA
5113    NA
4049    NA
7838    NA
3033    NA
4766    NA
5834    NA
10000 random_numbers.txt
```

In [12]:
```
!cp random_numbers.txt $HADOOP_HOME/input
!ls $HADOOP_HOME/input
```

```
capacity-scheduler.xml  hdfs-site.xml    kms-site.xml        yarn-
site.xml
core-site.xml           httpfs-site.xml  mapred-site.xml
hadoop-policy.xml       kms-acls.xml     random_numbers.txt
```

Hadoop streaming should automatically sort the keys. Since our integers are the keys, we can just reprint them using the mapper to sort them.

In [1]:
```
!hdfs namenode && hdfs datanode
!$HADOOP_HOME/sbin/start-yarn.sh

# CD to hadoop root dir
!cd $HADOOP_HOME
```

```
16/01/26 09:20:01 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ip-172-31-25-233.us-west-2.compute.interna
l/172.31.25.233
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 2.7.1
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/loca
l/hadoop/share/hadoop/common/lib/protobuf-java-2.5.0.jar:/usr/loca
l/hadoop/share/hadoop/common/lib/api-util-1.0.0-M20.jar:/usr/loca
l/hadoop/share/hadoop/common/lib/zookeeper-3.4.6.jar:/usr/local/ha
doop/share/hadoop/common/lib/hamcrest-core-1.3.jar:/usr/local/hado
op/share/hadoop/common/lib/slf4j-api-1.7.10.jar:/usr/local/hadoo
p/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/local/hadoop/sh
are/hadoop/common/lib/commons-httpclient-3.1.jar:/usr/local/hadoo
p/share/hadoop/common/lib/guava-11.0.2.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/jackson-core-asl-1.9.13.jar:/usr/local/hadoo
p/share/hadoop/common/lib/xz-1.0.jar:/usr/local/hadoop/share/hadoo
p/common/lib/java-xmlbuilder-0.4.jar:/usr/local/hadoop/share/hadoo
p/common/lib/servlet-api-2.5.jar:/usr/local/hadoop/share/hadoop/co
mmon/lib/commons-codec-1.4.jar:/usr/local/hadoop/share/hadoop/comm
on/lib/commons-collections-3.2.1.jar:/usr/local/hadoop/share/hadoo
p/common/lib/commons-digester-1.8.jar:/usr/local/hadoop/share/hado
op/common/lib/asm-3.2.jar:/usr/local/hadoop/share/hadoop/common/li
b/httpcore-4.2.5.jar:/usr/local/hadoop/share/hadoop/common/lib/com
mons-io-2.4.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-
configuration-1.6.jar:/usr/local/hadoop/share/hadoop/common/lib/co
mmons-net-3.1.jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-
log4j12-1.7.10.jar:/usr/local/hadoop/share/hadoop/common/lib/jets3
t-0.9.0.jar:/usr/local/hadoop/share/hadoop/common/lib/apacheds-ker
beros-codec-2.0.0-M15.jar:/usr/local/hadoop/share/hadoop/common/li
b/log4j-1.2.17.jar:/usr/local/hadoop/share/hadoop/common/lib/commo
ns-beanutils-core-1.8.0.jar:/usr/local/hadoop/share/hadoop/commo
n/lib/jersey-core-1.9.jar:/usr/local/hadoop/share/hadoop/common/li
b/commons-beanutils-1.7.0.jar:/usr/local/hadoop/share/hadoop/commo
n/lib/httpclient-4.2.5.jar:/usr/local/hadoop/share/hadoop/common/l
ib/jetty-6.1.26.jar:/usr/local/hadoop/share/hadoop/common/lib/jack
son-jaxrs-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/jax
b-impl-2.2.3-1.jar:/usr/local/hadoop/share/hadoop/common/lib/jetti
son-1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/jsch-0.1.4
2.jar:/usr/local/hadoop/share/hadoop/common/lib/jetty-util-6.1.2
6.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-client-
2.7.1.jar:/usr/local/hadoop/share/hadoop/common/lib/hadoop-annotat
ions-2.7.1.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-m
ath3-3.1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/junit-4.1
1.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-framework-
2.7.1.jar:/usr/local/hadoop/share/hadoop/common/lib/netty-3.6.2.Fi
nal.jar:/usr/local/hadoop/share/hadoop/common/lib/paranamer-2.3.ja
r:/usr/local/hadoop/share/hadoop/common/lib/activation-1.1.jar:/us
r/local/hadoop/share/hadoop/common/lib/commons-logging-1.1.3.ja
r:/usr/local/hadoop/share/hadoop/common/lib/snappy-java-1.0.4.1.ja
r:/usr/local/hadoop/share/hadoop/common/lib/jsp-api-2.1.jar:/usr/l
ocal/hadoop/share/hadoop/common/lib/xmlenc-0.52.jar:/usr/local/had
```

```
oop/share/hadoop/common/lib/commons-lang-2.6.jar:/usr/local/hadoo
p/share/hadoop/common/lib/gson-2.2.4.jar:/usr/local/hadoop/share/h
adoop/common/lib/avro-1.7.4.jar:/usr/local/hadoop/share/hadoop/com
mon/lib/api-asn1-api-1.0.0-M20.jar:/usr/local/hadoop/share/hadoo
p/common/lib/stax-api-1.0-2.jar:/usr/local/hadoop/share/hadoop/com
mon/lib/jersey-server-1.9.jar:/usr/local/hadoop/share/hadoop/commo
n/lib/curator-recipes-2.7.1.jar:/usr/local/hadoop/share/hadoop/com
mon/lib/jaxb-api-2.2.2.jar:/usr/local/hadoop/share/hadoop/common/l
ib/htrace-core-3.1.0-incubating.jar:/usr/local/hadoop/share/hadoo
p/common/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/apacheds-i18n-2.0.0-M15.jar:/usr/local/hadoo
p/share/hadoop/common/lib/mockito-all-1.8.5.jar:/usr/local/hadoo
p/share/hadoop/common/lib/jsr305-3.0.0.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/jersey-json-1.9.jar:/usr/local/hadoop/share/ha
doop/common/lib/commons-compress-1.4.1.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/jackson-xc-1.9.13.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/hadoop-auth-2.7.1.jar:/usr/local/hadoop/shar
e/hadoop/common/hadoop-common-2.7.1-tests.jar:/usr/local/hadoop/sh
are/hadoop/common/hadoop-nfs-2.7.1.jar:/usr/local/hadoop/share/had
oop/common/hadoop-common-2.7.1.jar:/usr/local/hadoop/share/hadoo
p/hdfs:/usr/local/hadoop/share/hadoop/hdfs/lib/protobuf-java-
2.5.0.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-cli-
1.2.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/guava-11.0.2.ja
r:/usr/local/hadoop/share/hadoop/hdfs/lib/jackson-core-asl-1.9.1
3.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/servlet-api-2.5.ja
r:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-codec-1.4.jar:/u
sr/local/hadoop/share/hadoop/hdfs/lib/xml-apis-1.3.04.jar:/usr/loc
al/hadoop/share/hadoop/hdfs/lib/asm-3.2.jar:/usr/local/hadoop/shar
e/hadoop/hdfs/lib/commons-io-2.4.jar:/usr/local/hadoop/share/hadoo
p/hdfs/lib/xercesImpl-2.9.1.jar:/usr/local/hadoop/share/hadoop/hdf
s/lib/commons-daemon-1.0.13.jar:/usr/local/hadoop/share/hadoop/hdf
s/lib/log4j-1.2.17.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jer
sey-core-1.9.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jetty-
6.1.26.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/netty-all-4.0.2
3.Final.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/jetty-util-
6.1.26.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/netty-3.6.2.Fin
al.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-logging-
1.1.3.jar:/usr/local/hadoop/share/hadoop/hdfs/lib/xmlenc-0.52.ja
r:/usr/local/hadoop/share/hadoop/hdfs/lib/commons-lang-2.6.jar:/us
r/local/hadoop/share/hadoop/hdfs/lib/jersey-server-1.9.jar:/usr/lo
cal/hadoop/share/hadoop/hdfs/lib/leveldbjni-all-1.8.jar:/usr/loca
l/hadoop/share/hadoop/hdfs/lib/htrace-core-3.1.0-incubating.jar:/u
sr/local/hadoop/share/hadoop/hdfs/lib/jackson-mapper-asl-1.9.13.ja
r:/usr/local/hadoop/share/hadoop/hdfs/lib/jsr305-3.0.0.jar:/usr/lo
cal/hadoop/share/hadoop/hdfs/hadoop-hdfs-2.7.1-tests.jar:/usr/loca
l/hadoop/share/hadoop/hdfs/hadoop-hdfs-nfs-2.7.1.jar:/usr/local/ha
doop/share/hadoop/hdfs/hadoop-hdfs-2.7.1.jar:/usr/local/hadoop/sha
re/hadoop/yarn/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/shar
e/hadoop/yarn/lib/aopalliance-1.0.jar:/usr/local/hadoop/share/hado
op/yarn/lib/zookeeper-3.4.6.jar:/usr/local/hadoop/share/hadoop/yar
n/lib/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/yarn/li
b/jersey-guice-1.9.jar:/usr/local/hadoop/share/hadoop/yarn/lib/gua
va-11.0.2.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jackson-cor
e-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/yarn/lib/xz-1.0.ja
```

```
r:/usr/local/hadoop/share/hadoop/yarn/lib/servlet-api-2.5.jar:/us
r/local/hadoop/share/hadoop/yarn/lib/guice-servlet-3.0.jar:/usr/lo
cal/hadoop/share/hadoop/yarn/lib/commons-codec-1.4.jar:/usr/loca
l/hadoop/share/hadoop/yarn/lib/commons-collections-3.2.1.jar:/us
r/local/hadoop/share/hadoop/yarn/lib/asm-3.2.jar:/usr/local/hadoo
p/share/hadoop/yarn/lib/commons-io-2.4.jar:/usr/local/hadoop/shar
e/hadoop/yarn/lib/zookeeper-3.4.6-tests.jar:/usr/local/hadoop/shar
e/hadoop/yarn/lib/log4j-1.2.17.jar:/usr/local/hadoop/share/hadoo
p/yarn/lib/jersey-core-1.9.jar:/usr/local/hadoop/share/hadoop/yar
n/lib/jetty-6.1.26.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jac
kson-jaxrs-1.9.13.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jax
b-impl-2.2.3-1.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jettiso
n-1.1.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jetty-util-6.1.2
6.jar:/usr/local/hadoop/share/hadoop/yarn/lib/netty-3.6.2.Final.ja
r:/usr/local/hadoop/share/hadoop/yarn/lib/activation-1.1.jar:/us
r/local/hadoop/share/hadoop/yarn/lib/commons-logging-1.1.3.jar:/us
r/local/hadoop/share/hadoop/yarn/lib/commons-lang-2.6.jar:/usr/loc
al/hadoop/share/hadoop/yarn/lib/stax-api-1.0-2.jar:/usr/local/hado
op/share/hadoop/yarn/lib/jersey-server-1.9.jar:/usr/local/hadoop/s
hare/hadoop/yarn/lib/leveldbjni-all-1.8.jar:/usr/local/hadoop/shar
e/hadoop/yarn/lib/javax.inject-1.jar:/usr/local/hadoop/share/hadoo
p/yarn/lib/jaxb-api-2.2.2.jar:/usr/local/hadoop/share/hadoop/yar
n/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/share/hadoo
p/yarn/lib/guice-3.0.jar:/usr/local/hadoop/share/hadoop/yarn/lib/j
sr305-3.0.0.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jersey-jso
n-1.9.jar:/usr/local/hadoop/share/hadoop/yarn/lib/commons-compres
s-1.4.1.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jackson-xc-
1.9.13.jar:/usr/local/hadoop/share/hadoop/yarn/lib/jersey-client-
1.9.jar:/usr/local/hadoop/share/hadoop/yarn/hadoop-yarn-applicatio
ns-distributedshell-2.7.1.jar:/usr/local/hadoop/share/hadoop/yar
n/hadoop-yarn-server-tests-2.7.1.jar:/usr/local/hadoop/share/hadoo
p/yarn/hadoop-yarn-server-web-proxy-2.7.1.jar:/usr/local/hadoop/sh
are/hadoop/yarn/hadoop-yarn-registry-2.7.1.jar:/usr/local/hadoop/s
hare/hadoop/yarn/hadoop-yarn-server-nodemanager-2.7.1.jar:/usr/loc
al/hadoop/share/hadoop/yarn/hadoop-yarn-server-resourcemanager-
2.7.1.jar:/usr/local/hadoop/share/hadoop/yarn/hadoop-yarn-server-s
haredcachemanager-2.7.1.jar:/usr/local/hadoop/share/hadoop/yarn/ha
doop-yarn-common-2.7.1.jar:/usr/local/hadoop/share/hadoop/yarn/had
oop-yarn-server-applicationhistoryservice-2.7.1.jar:/usr/local/had
oop/share/hadoop/yarn/hadoop-yarn-api-2.7.1.jar:/usr/local/hadoo
p/share/hadoop/yarn/hadoop-yarn-client-2.7.1.jar:/usr/local/hadoo
p/share/hadoop/yarn/hadoop-yarn-applications-unmanaged-am-launche
r-2.7.1.jar:/usr/local/hadoop/share/hadoop/yarn/hadoop-yarn-serve
r-common-2.7.1.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/pr
otobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/li
b/aopalliance-1.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/li
b/hamcrest-core-1.3.jar:/usr/local/hadoop/share/hadoop/mapreduce/l
ib/jersey-guice-1.9.jar:/usr/local/hadoop/share/hadoop/mapreduce/l
ib/jackson-core-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/mapr
educe/lib/xz-1.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/li
b/guice-servlet-3.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/l
ib/asm-3.2.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/common
s-io-2.4.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/log4j-
1.2.17.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/jersey-cor
```

```
e-1.9.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/hadoop-anno
tations-2.7.1.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/jun
it-4.11.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/netty-
3.6.2.Final.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/paran
amer-2.3.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/snappy-j
ava-1.0.4.1.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/avro-
1.7.4.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/jersey-serv
er-1.9.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/leveldbjn
i-all-1.8.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/javax.i
nject-1.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/jackson-m
apper-asl-1.9.13.jar:/usr/local/hadoop/share/hadoop/mapreduce/li
b/guice-3.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/lib/commo
ns-compress-1.4.1.jar:/usr/local/hadoop/share/hadoop/mapreduce/had
oop-mapreduce-examples-2.7.1.jar:/usr/local/hadoop/share/hadoop/ma
preduce/hadoop-mapreduce-client-core-2.7.1.jar:/usr/local/hadoop/s
hare/hadoop/mapreduce/hadoop-mapreduce-client-hs-2.7.1.jar:/usr/lo
cal/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-hs-plugi
ns-2.7.1.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapre
duce-client-app-2.7.1.jar:/usr/local/hadoop/share/hadoop/mapreduc
e/hadoop-mapreduce-client-jobclient-2.7.1-tests.jar:/usr/local/had
oop/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-
2.7.1.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduc
e-client-common-2.7.1.jar:/usr/local/hadoop/share/hadoop/mapreduc
e/hadoop-mapreduce-client-shuffle-2.7.1.jar:/usr/local/hadoop/cont
rib/capacity-scheduler/*.jar
STARTUP_MSG:   build = https://git-wip-us.apache.org/repos/asf/had
oop.git -r 15ecc87ccf4a0228f35af08fc56de536e6ce657a; compiled by
'jenkins' on 2015-06-29T06:04Z
STARTUP_MSG:   java = 1.7.0_91
************************************************************/
16/01/26 09:20:01 INFO namenode.NameNode: registered UNIX signal h
andlers for [TERM, HUP, INT]
16/01/26 09:20:01 INFO namenode.NameNode: createNameNode []
16/01/26 09:20:12 INFO impl.MetricsConfig: loaded properties from
hadoop-metrics2.properties
16/01/26 09:20:15 INFO impl.MetricsSystemImpl: Scheduled snapshot
period at 10 second(s).
16/01/26 09:20:15 INFO impl.MetricsSystemImpl: NameNode metrics sy
stem started
16/01/26 09:20:15 INFO namenode.NameNode: fs.defaultFS is hdfs://l
ocalhost:9000
16/01/26 09:20:15 INFO namenode.NameNode: Clients are to use local
host:9000 to access this namenode/service.
16/01/26 09:20:28 INFO hdfs.DFSUtil: Starting Web-server for hdfs
at: http://0.0.0.0:50070
16/01/26 09:20:31 INFO mortbay.log: Logging to org.slf4j.impl.Log4
jLoggerAdapter(org.mortbay.log) via org.mortbay.log.Slf4jLog
16/01/26 09:20:33 INFO server.AuthenticationFilter: Unable to init
ialize FileSignerSecretProvider, falling back to use random secret
s.
16/01/26 09:20:33 INFO http.HttpRequestLog: Http request log for h
ttp.requests.namenode is not defined
16/01/26 09:20:33 INFO http.HttpServer2: Added global filter 'safe
ty' (class=org.apache.hadoop.http.HttpServer2$QuotingInputFilter)
```

```
16/01/26 09:20:34 INFO http.HttpServer2: Added filter static_use
r_filter (class=org.apache.hadoop.http.lib.StaticUserWebFilter$Sta
ticUserFilter) to context hdfs
16/01/26 09:20:34 INFO http.HttpServer2: Added filter static_use
r_filter (class=org.apache.hadoop.http.lib.StaticUserWebFilter$Sta
ticUserFilter) to context logs
16/01/26 09:20:34 INFO http.HttpServer2: Added filter static_use
r_filter (class=org.apache.hadoop.http.lib.StaticUserWebFilter$Sta
ticUserFilter) to context static
16/01/26 09:20:36 INFO http.HttpServer2: Added filter 'org.apach
e.hadoop.hdfs.web.AuthFilter' (class=org.apache.hadoop.hdfs.web.Au
thFilter)
16/01/26 09:20:36 INFO http.HttpServer2: addJerseyResourcePackage:
packageName=org.apache.hadoop.hdfs.server.namenode.web.resources;o
rg.apache.hadoop.hdfs.web.resources, pathSpec=/webhdfs/v1/*
16/01/26 09:20:37 INFO http.HttpServer2: Jetty bound to port 50070
16/01/26 09:20:37 INFO mortbay.log: jetty-6.1.26
16/01/26 09:20:45 INFO mortbay.log: Started HttpServer2$SelectChan
nelConnectorWithSafeStartup@0.0.0.0:50070
16/01/26 09:20:47 WARN namenode.FSNamesystem: Only one image stora
ge directory (dfs.namenode.name.dir) configured. Beware of data lo
ss due to lack of redundant storage directories!
16/01/26 09:20:47 WARN namenode.FSNamesystem: Only one namespace e
dits storage directory (dfs.namenode.edits.dir) configured. Beware
of data loss due to lack of redundant storage directories!
16/01/26 09:20:49 INFO namenode.FSNamesystem: No KeyProvider foun
d.
16/01/26 09:20:49 INFO namenode.FSNamesystem: fsLock is fair:true
16/01/26 09:20:50 INFO blockmanagement.DatanodeManager: dfs.bloc
k.invalidate.limit=1000
16/01/26 09:20:50 INFO blockmanagement.DatanodeManager: dfs.nameno
de.datanode.registration.ip-hostname-check=true
16/01/26 09:20:50 INFO blockmanagement.BlockManager: dfs.namenod
e.startup.delay.block.deletion.sec is set to 000:00:00:00.000
16/01/26 09:20:50 INFO blockmanagement.BlockManager: The block del
etion will start around 2016 Jan 26 09:20:50
16/01/26 09:20:50 INFO util.GSet: Computing capacity for map Block
sMap
16/01/26 09:20:50 INFO util.GSet: VM type       = 64-bit
16/01/26 09:20:50 INFO util.GSet: 2.0% max memory 966.7 MB = 19.3
MB
16/01/26 09:20:50 INFO util.GSet: capacity      = 2^21 = 2097152 e
ntries
16/01/26 09:20:51 INFO blockmanagement.BlockManager: dfs.block.acc
ess.token.enable=false
16/01/26 09:20:51 INFO blockmanagement.BlockManager: defaultReplic
ation         = 1
16/01/26 09:20:51 INFO blockmanagement.BlockManager: maxReplicatio
n             = 512
16/01/26 09:20:51 INFO blockmanagement.BlockManager: minReplicatio
n             = 1
16/01/26 09:20:51 INFO blockmanagement.BlockManager: maxReplicatio
nStreams      = 2
16/01/26 09:20:51 INFO blockmanagement.BlockManager: shouldCheckFo
```

```
rEnoughRacks  = false
16/01/26 09:20:51 INFO blockmanagement.BlockManager: replicationRe
checkInterval = 3000
16/01/26 09:20:51 INFO blockmanagement.BlockManager: encryptDataTr
ansfer        = false
16/01/26 09:20:51 INFO blockmanagement.BlockManager: maxNumBlocksT
oLog          = 1000
16/01/26 09:20:51 INFO namenode.FSNamesystem: fsOwner
= ubuntu (auth:SIMPLE)
16/01/26 09:20:51 INFO namenode.FSNamesystem: supergroup
= supergroup
16/01/26 09:20:51 INFO namenode.FSNamesystem: isPermissionEnabled
= true
16/01/26 09:20:51 INFO namenode.FSNamesystem: HA Enabled: false
16/01/26 09:20:51 INFO namenode.FSNamesystem: Append Enabled: true
16/01/26 09:20:52 INFO util.GSet: Computing capacity for map INode
Map
16/01/26 09:20:52 INFO util.GSet: VM type       = 64-bit
16/01/26 09:20:52 INFO util.GSet: 1.0% max memory 966.7 MB = 9.7 M
B
16/01/26 09:20:52 INFO util.GSet: capacity      = 2^20 = 1048576 e
ntries
16/01/26 09:20:52 INFO namenode.FSDirectory: ACLs enabled? false
16/01/26 09:20:52 INFO namenode.FSDirectory: XAttrs enabled? true
16/01/26 09:20:52 INFO namenode.FSDirectory: Maximum size of an xa
ttr: 16384
16/01/26 09:20:52 INFO namenode.NameNode: Caching file names occur
ing more than 10 times
16/01/26 09:20:53 INFO util.GSet: Computing capacity for map cache
dBlocks
16/01/26 09:20:53 INFO util.GSet: VM type       = 64-bit
16/01/26 09:20:53 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4
MB
16/01/26 09:20:53 INFO util.GSet: capacity      = 2^18 = 262144 en
tries
16/01/26 09:20:53 INFO namenode.FSNamesystem: dfs.namenode.safemod
e.threshold-pct = 0.9990000128746033
16/01/26 09:20:53 INFO namenode.FSNamesystem: dfs.namenode.safemod
e.min.datanodes = 0
16/01/26 09:20:53 INFO namenode.FSNamesystem: dfs.namenode.safemod
e.extension     = 30000
16/01/26 09:20:53 INFO metrics.TopMetrics: NNTop conf: dfs.namenod
e.top.window.num.buckets = 10
16/01/26 09:20:53 INFO metrics.TopMetrics: NNTop conf: dfs.namenod
e.top.num.users = 10
16/01/26 09:20:53 INFO metrics.TopMetrics: NNTop conf: dfs.namenod
e.top.windows.minutes = 1,5,25
16/01/26 09:20:53 INFO namenode.FSNamesystem: Retry cache on namen
ode is enabled
16/01/26 09:20:53 INFO namenode.FSNamesystem: Retry cache will use
0.03 of total heap and retry cache entry expiry time is 600000 mil
lis
16/01/26 09:20:53 INFO util.GSet: Computing capacity for map NameN
odeRetryCache
```

```
16/01/26 09:20:53 INFO util.GSet: VM type       = 64-bit
16/01/26 09:20:53 INFO util.GSet: 0.029999999329447746% max memory
966.7 MB = 297.0 KB
16/01/26 09:20:53 INFO util.GSet: capacity      = 2^15 = 32768 ent
ries
16/01/26 09:20:53 INFO common.Storage: Lock on /usr/local/hadoop/h
adoop_data/hdfs/namenode/current/in_use.lock acquired by nodename
9106@ip-172-31-25-233.us-west-2.compute.internal
16/01/26 09:20:58 INFO namenode.FileJournalManager: Recovering unf
inalized segments in /usr/local/hadoop/hadoop_data/hdfs/namenode/c
urrent/current
16/01/26 09:21:01 INFO namenode.FileJournalManager: Finalizing edi
ts file /usr/local/hadoop/hadoop_data/hdfs/namenode/current/curren
t/edits_inprogress_0000000000000000069 -> /usr/local/hadoop/hadoo
p_data/hdfs/namenode/current/current/edits_0000000000000000069-000
0000000000000069
16/01/26 09:21:04 INFO namenode.FSImageFormatPBINode: Loading 1 IN
odes.
16/01/26 09:21:05 INFO namenode.FSImageFormatProtobuf: Loaded FSIm
age in 3 seconds.
16/01/26 09:21:05 INFO namenode.FSImage: Loaded image for txid 0 f
rom /usr/local/hadoop/hadoop_data/hdfs/namenode/current/current/fs
image_0000000000000000000
16/01/26 09:21:05 INFO namenode.FSImage: Reading org.apache.hadoo
p.hdfs.server.namenode.RedundantEditLogInputStream@3749dde5 expect
ing start txid #1
16/01/26 09:21:05 INFO namenode.FSImage: Start loading edits file
/usr/local/hadoop/hadoop_data/hdfs/namenode/current/current/edit
s_0000000000000000001-0000000000000000068
16/01/26 09:21:05 INFO namenode.EditLogInputStream: Fast-forwardin
g stream '/usr/local/hadoop/hadoop_data/hdfs/namenode/current/curr
ent/edits_0000000000000000001-0000000000000000068' to transaction
ID 1
16/01/26 09:21:07 INFO namenode.FSEditLogLoader: replaying edit lo
g: 2/68 transactions completed. (3%)
16/01/26 09:21:09 INFO namenode.FSEditLogLoader: replaying edit lo
g: 26/68 transactions completed. (38%)
16/01/26 09:21:09 INFO namenode.FSImage: Edits file /usr/local/had
oop/hadoop_data/hdfs/namenode/current/current/edits_00000000000000
00001-0000000000000000068 of size 1048576 edits # 68 loaded in 3 s
econds
16/01/26 09:21:09 INFO namenode.FSImage: Reading org.apache.hadoo
p.hdfs.server.namenode.RedundantEditLogInputStream@528701be expect
ing start txid #69
16/01/26 09:21:09 INFO namenode.FSImage: Start loading edits file
/usr/local/hadoop/hadoop_data/hdfs/namenode/current/current/edit
s_0000000000000000069-0000000000000000069
16/01/26 09:21:09 INFO namenode.EditLogInputStream: Fast-forwardin
g stream '/usr/local/hadoop/hadoop_data/hdfs/namenode/current/curr
ent/edits_0000000000000000069-0000000000000000069' to transaction
ID 1
16/01/26 09:21:09 INFO namenode.FSImage: Edits file /usr/local/had
oop/hadoop_data/hdfs/namenode/current/current/edits_00000000000000
00069-0000000000000000069 of size 1048576 edits # 1 loaded in 0 se
```

```
conds
16/01/26 09:21:09 INFO namenode.FSNamesystem: Need to save fs imag
e? true (staleImage=true, haEnabled=false, isRollingUpgrade=false)
16/01/26 09:21:09 INFO namenode.FSImage: Save namespace ...
16/01/26 09:21:10 INFO namenode.NNStorageRetentionManager: Going t
o retain 2 images with txid >= 0
16/01/26 09:21:10 INFO namenode.FSEditLog: Starting log segment at
70
16/01/26 09:21:11 INFO namenode.NameCache: initialized with 0 entr
ies 0 lookups
16/01/26 09:21:11 INFO namenode.FSNamesystem: Finished loading FSI
mage in 18383 msecs
16/01/26 09:21:16 INFO namenode.NameNode: RPC server is binding to
localhost:9000
16/01/26 09:21:16 INFO ipc.CallQueueManager: Using callQueue class
java.util.concurrent.LinkedBlockingQueue
16/01/26 09:21:16 INFO ipc.Server: Starting Socket Reader #1 for p
ort 9000
16/01/26 09:21:18 INFO namenode.FSNamesystem: Registered FSNamesys
temState MBean
16/01/26 09:21:19 INFO namenode.LeaseManager: Number of blocks und
er construction: 0
16/01/26 09:21:19 INFO namenode.LeaseManager: Number of blocks und
er construction: 0
16/01/26 09:21:19 INFO hdfs.StateChange: STATE* Safe mode ON.
The reported blocks 0 needs additional 6 blocks to reach the thres
hold 0.9990 of total blocks 6.
The number of live datanodes 0 has reached the minimum number 0. S
afe mode will be turned off automatically once the thresholds have
been reached.
16/01/26 09:21:19 INFO blockmanagement.DatanodeDescriptor: Number
of failed storage changes from 0 to 0
16/01/26 09:21:20 INFO ipc.Server: IPC Server Responder: starting
16/01/26 09:21:20 INFO namenode.NameNode: NameNode RPC up at: loca
lhost/127.0.0.1:9000
16/01/26 09:21:20 INFO namenode.FSNamesystem: Starting services re
quired for active state
16/01/26 09:21:20 INFO ipc.Server: IPC Server listener on 9000: st
arting
16/01/26 09:21:20 INFO blockmanagement.CacheReplicationMonitor: St
arting CacheReplicationMonitor with interval 30000 milliseconds
^C16/01/26 09:51:01 ERROR namenode.NameNode: RECEIVED SIGNAL 2: SI
GINT
16/01/26 09:51:01 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ip-172-31-25-233.us-west-
2.compute.internal/172.31.25.233
************************************************************/

starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-u
buntu-resourcemanager-ip-172-31-25-233.out
localhost: starting nodemanager, logging to /usr/local/hadoop/log
s/yarn-ubuntu-nodemanager-ip-172-31-25-233.out
```

In [29]:
```
# check hdfs status and yarn nodes
!hdfs dfsadmin -report
!yarn node -list
```

```
Configured Capacity: 0 (0 B)
Present Capacity: 0 (0 B)
DFS Remaining: 0 (0 B)
DFS Used: 0 (0 B)
DFS Used%: NaN%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
16/01/25 15:47:32 INFO client.RMProxy: Connecting to ResourceManag
er at /0.0.0.0:8032
Total Nodes:1
         Node-Id             Node-State Node-Http-Address       Nu
mber-of-Running-Containers
ec2-54-213-63-253.us-west-2.compute.amazonaws.com:35267         RU
NNING   ec2-54-213-63-253.us-west-2.compute.amazonaws.com:8042
0
```

For HDFS health:

http:/ec2-54-213-63-253.us-west-2.compute.amazonaws.com:50070/dfshealth.html#tab-overview (http:/ec2-54-213-63-253.us-west-2.compute.amazonaws.com:50070/dfshealth.html#tab-overview)

For YARN cluster/job manager: http://ec2-54-213-63-253.us-west-2.compute.amazonaws.com:8088/cluster (http://ec2-54-213-63-253.us-west-2.compute.amazonaws.com:8088/cluster)

In [21]:
```
%%writefile mapper.py
#!/usr/bin/python

import sys

# Simply read and print out key value pairs from input
# note, writing to standard out seems more secure than print
for line in sys.stdin:
    sys.stdout.write(line)
    #key, empty_string = line.split('\t')
    #print str(key) +"\t" + empty_string
```

```
Overwriting mapper.py
```

In [18]:
```
%%writefile reducer.py
#!/usr/bin/python

import sys

# Simply read and print out key value pairs from input
for line in sys.stdin:
    #key, empty_string = line.split('\t')
    #print str(key) +"\t" + empty_string
    sys.stdout.write(line)
```

```
Overwriting reducer.py
```

In [22]:
```
!chmod a+x mapper.py
!chmod a+x reducer.py
```

In [28]:
```
!./mapper.py <random_numbers.txt| sort -n | ./reducer.py >rand_nu
m.out
!head rand_num.out
!tail rand_num.out
```

```
0       NA
1       NA
1       NA
2       NA
5       NA
6       NA
9       NA
10      NA
10      NA
11      NA
9989    NA
9989    NA
9992    NA
9995    NA
9995    NA
9997    NA
9999    NA
9999    NA
9999    NA
10000   NA
```

In [2]:
```
!$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-*streaming*.jar -m
apper $HADOOP_HOME/mapper.py -reducer reducer.py -input /user/safyr
e/input/random_numbers.txt -output /user/safyre/output/SortOutput
```

```
packageJobJar: [/tmp/hadoop-unjar1215359286937913206/] [] /tmp/str
eamjob4040146808467195045.jar tmpDir=null
16/01/26 04:23:46 INFO client.RMProxy: Connecting to ResourceManag
er at /0.0.0.0:8032
16/01/26 04:23:52 INFO client.RMProxy: Connecting to ResourceManag
er at /0.0.0.0:8032
16/01/26 04:24:02 INFO mapred.FileInputFormat: Total input paths t
o process : 1
16/01/26 04:24:04 INFO mapreduce.JobSubmitter: number of splits:2
16/01/26 04:24:13 INFO mapreduce.JobSubmitter: Submitting tokens f
or job: job_1453781403055_0002
16/01/26 04:24:23 INFO impl.YarnClientImpl: Submitted application
application_1453781403055_0002
16/01/26 04:24:27 INFO mapreduce.Job: The url to track the job: ht
tp://ip-172-31-25-233:8088/proxy/application_1453781403055_0002/
16/01/26 04:24:27 INFO mapreduce.Job: Running job: job_14537814030
55_0002
16/01/26 04:27:48 INFO mapreduce.Job: Job job_1453781403055_0002 r
unning in uber mode : false
16/01/26 04:27:48 INFO mapreduce.Job:  map 0% reduce 0%
16/01/26 04:30:09 INFO ipc.Client: Retrying connect to server: ip-
172-31-25-233.us-west-2.compute.internal/172.31.25.233:50272. Alre
ady tried 0 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=3, sleepTime=1000 MILLISECONDS)
16/01/26 04:30:10 INFO ipc.Client: Retrying connect to server: ip-
172-31-25-233.us-west-2.compute.internal/172.31.25.233:50272. Alre
ady tried 1 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=3, sleepTime=1000 MILLISECONDS)
16/01/26 04:30:11 INFO ipc.Client: Retrying connect to server: ip-
172-31-25-233.us-west-2.compute.internal/172.31.25.233:50272. Alre
ady tried 2 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=3, sleepTime=1000 MILLISECONDS)
16/01/26 04:30:12 INFO ipc.Client: Retrying connect to server: ip-
172-31-25-233.us-west-2.compute.internal/172.31.25.233:50272. Alre
ady tried 0 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=3, sleepTime=1000 MILLISECONDS)
16/01/26 04:30:13 INFO ipc.Client: Retrying connect to server: ip-
172-31-25-233.us-west-2.compute.internal/172.31.25.233:50272. Alre
ady tried 1 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=3, sleepTime=1000 MILLISECONDS)
16/01/26 04:30:14 INFO ipc.Client: Retrying connect to server: ip-
172-31-25-233.us-west-2.compute.internal/172.31.25.233:50272. Alre
ady tried 2 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=3, sleepTime=1000 MILLISECONDS)
16/01/26 04:31:54 INFO mapreduce.Job: Job job_1453781403055_0002 f
ailed with state FAILED due to: Application application_1453781403
055_0002 failed 2 times due to AM Container for appattempt_1453781
403055_0002_000002 exited with  exitCode: 1
For more detailed output, check application tracking page:http://i
p-172-31-25-233:8088/cluster/app/application_1453781403055_0002The
n, click on links to logs of each attempt.
Diagnostics: Exception from container-launch.
Container id: container_1453781403055_0002_02_000001
```

```
Exit code: 1
Stack trace: ExitCodeException exitCode=1:
        at org.apache.hadoop.util.Shell.runCommand(Shell.java:545)
        at org.apache.hadoop.util.Shell.run(Shell.java:456)
        at org.apache.hadoop.util.Shell$ShellCommandExecutor.execu
te(Shell.java:722)
        at org.apache.hadoop.yarn.server.nodemanager.DefaultContai
nerExecutor.launchContainer(DefaultContainerExecutor.java:211)
        at org.apache.hadoop.yarn.server.nodemanager.containermana
ger.launcher.ContainerLaunch.call(ContainerLaunch.java:302)
        at org.apache.hadoop.yarn.server.nodemanager.containermana
ger.launcher.ContainerLaunch.call(ContainerLaunch.java:82)
        at java.util.concurrent.FutureTask.run(FutureTask.java:26
2)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(Threa
dPoolExecutor.java:1145)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(Thre
adPoolExecutor.java:615)
        at java.lang.Thread.run(Thread.java:745)


Container exited with a non-zero exit code 1
Failing this attempt. Failing the application.
16/01/26 04:31:56 INFO mapreduce.Job: Counters: 0
16/01/26 04:31:56 ERROR streaming.StreamJob: Job not successful!
Streaming Command Failed!
```

HW2.2. WORDCOUNT

Using the Enron data from HW1 and Hadoop MapReduce streaming, write the mapper/reducer job that will determine the word count (number of occurrences) of each white-space delimitted token (assume spaces, fullstops, comma as delimiters). Examine the word “assistance” and report its word count results.

CROSSCHECK: `>grep assistance enronemail_1h.txt|cut -d$'\t' -f4| grep assistance|wc -l`

```
`8`
```

#NOTE "assistance" occurs on 8 lines but how many times does the token occur? 10 times! This is the number we are looking for!

In [62]:
```
%%writefile mapper.py
#!/usr/bin/python
import re, string
import sys
import os
import numpy as np

# store a regex expression into a pattern object
# that seeks words including underscores and single quotes
WORD_RE = re.compile(r"[\w']+")
translate_table = string.maketrans("","") #empty translation

# file input
filename = sys.argv[1]

# list of words argument, '*' means all words
word_list = sys.argv[2]
count = 0

with open(filename, 'rU') as f:
    for line in f.readlines():
        #strip punctuation from line
        line = line.translate(translate_table, string.punctuation)

        # if not all words selected,
        # go through each word in word list and count occurances
        if word_list != '*':
            for word in word_list.split():
                counts = [int(1) if (x == word) and (word.isalph
a()) else int(0) for x in WORD_RE.findall(line)]
                counts = np.array(counts)

                if counts.sum() > 0:
                    print word + "\t" + str(counts.sum())
        else:
            for word in line.split():
                if word.isalpha():
                    print word + "\t"+ str(1)
```

```
Overwriting mapper.py
```

In [54]:
```
%%writefile reducer.py
#!/usr/bin/python

# recycled code from W205, sorry
import re, string
import sys
import os
import numpy as np

#if a new word enters the fray, print the current word and its coun
ts
def wcount(prev_word ,counts):
    if prev_word is not None:
            print(prev_word + "\t" + str(counts))

prev_word = None
counts = 0

for line in sys.stdin:
    word, value =line.split("\t",1)
    if word != prev_word:
        wcount(prev_word, counts)
        prev_word = word
        counts = 0
    counts += eval(value)

# A print just for the final word
wcount(prev_word, counts)
```

```
Overwriting reducer.py
```

In [63]:
```
!chmod a+x mapper.py
!chmod a+x reducer.py
!./mapper.py enronemail_1h.txt assistance |sort| ./reducer.py
```

```
assistance      10
```

HW2.2.1

Using Hadoop MapReduce and your wordcount job (from HW2.2) determine the top-10 occurring tokens (most frequent tokens)

In [69]:
```
!./mapper.py enronemail_1h.txt \* |sort | ./reducer.py |sort -k 2n
>hw221.out
```

In [71]:
```
# sorted in ascending order
!tail hw221.out
```

```
this    260
for     373
your    391
in      415
you     427
a       529
of      560
and     662
to      961
the     1246
```

HW2.3. Multinomial NAIVE BAYES with NO Smoothing

Using the Enron data from HW1 and Hadoop MapReduce, write a mapper/reducer job(s) that will both learn Naive Bayes classifier and classify the Enron email messages using the learnt Naive Bayes classifier. Use all white-space delimitted tokens as independent input variables (assume spaces, fullstops, commas as delimiters). Note: for multinomial Naive Bayes, the Pr(X="assistance"|Y=SPAM) is calculated as follows:

`the number of times "assistance" occurs in SPAM labeled documents / the number of words in documents labeled SPAM`

E.g., "assistance" occurs 5 times in all of the documents Labeled SPAM, and the length in terms of the number of words in all documents labeled as SPAM (when concatenated) is 1,000. Then $Pr(X = \text{"}assistance\text{"}|Y = SPAM) = 5/1000$*. Note this is a multinomial estimation of the class conditional for a Naive Bayes Classifier. No smoothing is needed in this HW. Multiplying lots of probabilities, which are between 0 and 1, can result in floating-point underflow. Since* $log(xy) = log(x) + log(y)$*, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities. Please pay attention to probabilites that are zero! They will need special attention. Count up how many times you need to process a zero probabilty for each class and report.*

Report the performance of your learnt classifier in terms of misclassifcation error rate of your multinomial Naive Bayes Classifier. Plot a histogram of the posterior probabilities (i.e., Pr(Class|Doc)) for each class over the training set. Summarize what you see.

Error Rate = misclassification rate with respect to a provided set (say training set in this case). It is more formally defined here:

Let DF represent the evalution set in the following:
$Err(Model, DF) = |(X, c(X)) \in DF : c(X) != Model(x)|/|DF|$

Where || denotes set cardinality; c(X) denotes the class of the tuple X in DF; and Model(X) denotes the class inferred by the Model "Model"

In [121]:
```
%%writefile mapper.py
#!/usr/bin/python
import re, string
import sys
import os
import numpy as np

# store a regex expression into a pattern object
# that seeks words including underscores and single quotes
WORD_RE = re.compile(r"[\w']+")
TRUTH_RE = re.compile(r"\t(\d)\t")
translate_table = string.maketrans("","") #empty translation


# file input
filename = sys.argv[1]

# for this part, just assume word_list is length 1
word_list = sys.argv[2]

# Avoid KeyError if no data in chunk
#counts_dict = dict.fromkeys(['0', '1'], 0)
counts_dict = {}
doc_len    = 0
spam_count = 0
ham_count  = 0

line_count = 0
with open(filename, 'rU') as f:
    for line in f.readlines():

        # Parse out TRUTH
        # truth is the actual label provided in the data
        # 1 = spam, 0 = ham
        #key is the id of the emai
        key = line.split()[0]
        try:
            truth = TRUTH_RE.findall(line)[0]

        except:
            #for some reason line 59 gives problems
            # truth = '1'
            continue

        # Remove punctuation
        line = line.translate(translate_table, string.punctuation)

        '''
         # define empty dictionaries
        for category in ['0','1']:
            counts_dict[category] = {}
        '''
```

```
        if word_list != "*":
            for word in word_list.split():
                #doc_len = len(line.split())
                counts = [1 if (x == word) and (x.isalpha()) else 0
for x in WORD_RE.findall(line)]
                counts = np.array(counts)

                # Only pass to reducer if the word is present
                if counts.sum() > 0:
                    count = counts.sum()
                    #counts_dict[truth][word] = counts_dict[trut
h].get(word, 0) + int(count)
                    print key + "\t" + truth + "\t" + word + "\t" +
str(count) #+ "\t" + str(doc_len)

        else:
            for word in list(set(line.split())):
                counts = [1 if (x == word) and (x.isalpha()) else 0
for x in WORD_RE.findall(line)]
                counts = np.array(counts)

                count = counts.sum()
                #counts_dict[truth][word] = counts_dict[truth].ge
t(word, 0) + int(count)
                print key +"\t" + truth + "\t" + word + "\t" + st
r(count) #+ "\t" + str(doc_len)

'''
for category, word_dictionary in counts_dict.iteritems():
    for words, count in counts_dict[category].iteritems():
        print key + category + "\t" + words + "\t" + str(count) +
"\t" + str(doc_len)
'''
```

```
Overwriting mapper.py
```

In [164]:
```
%%writefile reducer.py
#!/usr/bin/python
import re, sys
import numpy as np
from math import log

## training, gather all the counts and calculate corpus-wide prior
s, etc
## data come in as strings,
## ID TRUTH WORD COUNT
# define empty dictionaries


# number of documents in each class
N_spam_docs = 0
N_ham_docs  = 0

# number of terms in each class
N_spam_terms = 0
N_ham_terms = 0
counts_dict ={}
keys_dict = {}

for line in sys.stdin:
    key, truth, word, count = line.split()
    keys_dict[key+"_"+truth] = {}
    keys_dict[key+"_"+truth][word] = {}

    for i in ['0', '1']:
        counts_dict[i] = {}
    # tabulate word counts for each class
    counts_dict[truth][word] = counts_dict[truth].get(word, 0) + in
t(count)

    if truth == '1':
        N_spam_docs  += 1
        N_spam_terms += int(count)
    else:
        N_ham_docs   += 1
        N_ham_terms += int(count)

priors = {'0': float(N_ham_docs)/(N_spam_docs+N_ham_docs),
          '1': float(N_spam_docs)/(N_spam_docs+N_ham_docs)}

prior_counts = {'0': float(N_ham_terms),
          '1': float(N_spam_terms)}

## Calculate conditional probabilities
## P(word | class)
posteriors = {}
for category in ['0', '1']:
    posteriors[category] = {}
    for word in counts_dict[category].keys():
```

```
        posteriors[category][word] = float(counts_dict[category][wo
rd])/float(prior_counts[category])

for category in ['0', '1']:
    for key, word in keys_dict.iteritems():
        print key.split('_')[0] + "\t" +key.split('_')[1] + "\t" +
str(word) + '\t' +str(priors['0']) + \
        '\t' + str(priors['1']) + '\t' + str(posteriors['0'][word])
+ '\t' + str(posteriors['1'][word])
        #keys_dict[key][word]['prior_ham']    = priors['0']
        #keys_dict[key][word]['prior_spam']   = priors['1']
        #keys_dict[key][word]['posterior_ham'] = posteriors['0'][wo
rd]
        #keys_dict[key][word]['posterior_spam'] = posteriors['1'][w
ord]




#print "Priors are: "
#for category in priors:
#    print category + " " + str(priors[category]) + "n = " +str(pri
or_counts[category])

spam_vocab = counts_dict['1'].keys()
ham_vocab  = counts_dict['0'].keys()

spam_vocab_n = len(counts_dict['1'].keys())
ham_vocab_n  = len(counts_dict['0'].keys())

# all unique words from both classes
vocab = list(set(counts_dict['0'].keys()).union(counts_dict['1'].ke
ys()))
len_vocab = len(vocab)

#print "\nPosteriors are: "
#for category in posteriors:
#    for word in posteriors[category]:
#        print word + " in class " + category + " " + str(posterior
s[category][word]) + "\n"
```

```
Overwriting reducer.py
```

In [165]:
```
%%writefile reducer2.py

## Testing the classifer
## Without laplacian transform
#print "DOC_ID | TRUTH | CLASS "
#print "======================\n"


def cum_log_probs(prev_key, label, prediction):# ham_score, spam_sc
ore):
    if prev_key is not None:
        print prev_key + "\t" + label + "\t" +str(prediction)# +  "
" +ham_score + " " + spam_score

prev_key = None
counts = 0
prediction =0
correct = 0

'''
for line in sys.stdin:
    word, value =line.split("\t",1)
    if word!=prev_word:
        wcount(prev_word, counts)
        prev_word = word
        counts = 0
    counts += eval(value)

wcount(prev_word, counts)
'''

# This could probably be another MapReduce job...
for line in sys.stdin:
    for key, truth, word, prior_ham, prior_spam, posterior_ham, pos
terior_spam in line.split():
        print key, truth, word, count
        if key != prev_key:
            # Dump the previous key's statistics
            if prev_key is not None:
                if int(prediction) == int(truth):
                    correct +=1
                #print prev_key + "\t" + truth + "\t" +str(predicti
on) +  " " +str(score[0]) + " " + str(score[1])
                cum_logs_prob(prev_key, truth, prediction)# str(sco
re[0]), str(score[1]))
            #initialize scores for new sample
            score = [0,0]
            score[0]= log(prior_ham) + log(float(posterior_ham))
            score[1] = log(prior_spam) + log(float(posterior_spam))
            #for category in ['0', '1']:
            #    idx = int(category)
            #    score[idx] =  log(priors[category])
                   #score[idx] += log(float(posteriors[category][w
```

```
ord]))

            prev_key = key

        else:
            score[0] += log(float(posterior_ham))
            score[1] += log(float(posterior_spam))
            #for category in ['0', '1']:
            #    for word in posteriors[category]:
            #        score[idx] += log(float(posteriors[category][w
ord]))

            score = np.array(score)
            prediction = score.argmax()

#print prev_key + "\t" + truth + "\t" +str(prediction) +  " " +st
r(score[0]) + " " + str(score[1])
cum_log_probs(prev_key, truth, prediction)# str(score[0]), str(scor
e[1]))

accuracy = float(correct)/(N_spam_docs+N_ham_docs)*100.0
print "Accuracy: ", accuracy
```

```
Overwriting reducer2.py
```

In [166]:
```
!chmod a+x mapper.py
!chmod a+x reducer.py
!chmod a+x reducer2.py
!./mapper.py enronemail_1h.txt assistance | sort | ./reducer.py
|./reducer2.py
```

```
./reducer2.py: 8: ./reducer2.py: Syntax error: "(" unexpected
Traceback (most recent call last):
  File "./reducer.py", line 56, in <module>
    '\t' + str(priors['1']) + '\t' + str(posteriors['0'][word]) +
'\t' + str(posteriors['1'][word])
TypeError: unhashable type: 'dict'
```

HW2.4

Repeat HW2.3 with the following modification: use Laplace plus-one smoothing. Compare the misclassifcation error rates for 2.3 versus 2.4 and explain the differences.

For a quick reference on the construction of the Multinomial NAIVE BAYES classifier that you will code, please consult the "Document Classification" section of the following wikipedia page:

https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Document_classification (https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Document_classification)

OR the original paper by the curators of the Enron email data:

http://www.aueb.gr/users/ion/docs/ceas2006_paper.pdf (http://www.aueb.gr/users/ion/docs/ceas2006_paper.pdf)

HW2.5.

Repeat HW2.4. This time when modeling and classification ignore tokens with a frequency of less than three (3) in the training set. How does it affect the misclassifcation error of learnt naive multinomial Bayesian Classifier on the training dataset:

HW2.6

Benchmark your code with the Python SciKit-Learn implementation of the multinomial Naive Bayes algorithm

It always a good idea to benchmark your solutions against publicly available libraries such as SciKit-Learn. The Machine Learning toolkit available in Python. In this exercise, we benchmark ourselves against the SciKit-Learn implementation of multinomial Naive Bayes.
For more information on this implementation see: http://scikit-learn.org/stable/modules/naive_bayes.html (http://scikit-learn.org/stable/modules/naive_bayes.html)

In this exercise, please complete the following:

- *Run the Multinomial Naive Bayes algorithm (using default settings) from SciKit-Learn over the same training data used in HW2.5 and report the misclassification error (please note some data preparation might be needed to get the Multinomial Naive Bayes algorithm from SkiKit-Learn to run over this dataset)*
- *Prepare a table to present your results, where rows correspond to approach used (SkiKit-Learn versus your Hadoop implementation) and the column presents the training misclassification error*
- *Explain/justify any differences in terms of training error rates over the dataset in HW2.5 between your Multinomial Naive Bayes implementation (in Map Reduce) versus the Multinomial Naive Bayes implementation in SciKit-Learn*

In [157]:
```
%%writefile sklearn_run.py
#!/usr/bin/env python

import sys
import os
import numpy as np
import pandas as pd
import re

from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import *


data = pd.read_csv("enronemail_1h.txt",sep='\t',header=None)
data.columns = ['ID', 'TRUTH', 'SUBJECT', 'TEXT']
data = data.replace(np.nan,' ', regex=True)
train_data , train_labels = data['SUBJECT']+data['TEXT'] , data['TR
UTH']
vec = CountVectorizer()
vec_t = vec.fit_transform(train_data)

#Fit and predict Naivebayes
clf = MultinomialNB(alpha = 1.0)
clf.fit(vec_t, train_labels)
y_pred = clf.predict(vec_t)
err = 1- metrics.accuracy_score(train_labels, y_pred)
print "Training Error: " + str(err)
```

```
Overwriting sklearn_run.py
```

In [158]:
```
!chmod a+x sklearn_run.py
!./sklearn_run.py
```

```
Traceback (most recent call last):
  File "./sklearn_run.py", line 24, in <module>
    clf.fit(vec_t, train_labels)
  File "/home/ubuntu/anaconda2/lib/python2.7/site-packages/sklear
n/naive_bayes.py", line 531, in fit
    Y = labelbin.fit_transform(y)
  File "/home/ubuntu/anaconda2/lib/python2.7/site-packages/sklear
n/base.py", line 455, in fit_transform
    return self.fit(X, **fit_params).transform(X)
  File "/home/ubuntu/anaconda2/lib/python2.7/site-packages/sklear
n/preprocessing/label.py", line 308, in fit
    self.classes_ = unique_labels(y)
  File "/home/ubuntu/anaconda2/lib/python2.7/site-packages/sklear
n/utils/multiclass.py", line 99, in unique_labels
    raise ValueError("Unknown label type: %s" % repr(ys))
ValueError: Unknown label type: (array([0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0,
       0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0,
       1.0, 1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
1.0,
       1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0,
0.0,
       0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 1.0, ' ', 1.0, 1.0, 1.0, 0.0,
0.0,
       0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0,
1.0,
       1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0,
1.0,
       1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 1.0, 1.0], dtype=ob
ject),)
```

HW 2.6.1 OPTIONAL (note this exercise is a stretch HW and optional)

- *Run the Bernoulli Naive Bayes algorithm from SciKit-Learn (using default settings) over the same training data used in HW2.6 and report the misclassification error*
- *Discuss the performance differences in terms of misclassification error rates over the dataset in HW2.5 between the Multinomial Naive Bayes implementation in SciKit-Learn with the Bernoulli Naive Bayes implementation in SciKit-Learn. Why such big differences. Explain.*

Which approach to Naive Bayes would you recommend for SPAM detection? Justify your selection.

HW2.7 OPTIONAL (note this exercise is a stretch HW and optional)

The Enron SPAM data in the following folder enron1-Training-Data-RAW is in raw text form (with subfolders for SPAM and HAM that contain raw email messages in the following form:

- Line 1 contains the subject
- The remaining lines contain the body of the email message.

In Python write a script to produce a TSV file called train-Enron-1.txt that has a similar format as the enronemail_1h.txt that you have been using so far. Please pay attend to funky characters and tabs. Check your resulting formated email data in Excel and in Python (e.g., count up the number of fields in each row; the number of SPAM mails and the number of HAM emails). Does each row correspond to an email record with four values? Note: use "NA" to denote empty field values.

HW2.8 OPTIONAL

Using Hadoop Map-Reduce write job(s) to perform the following:

- *Train a multinomial Naive Bayes Classifier with Laplace plus one smoothing using the data extracted in HW2.7 (i.e., train-Enron-1.txt). Use all white-space delimitted tokens as independent input variables (assume spaces, fullstops, commas as delimiters). Drop tokens with a frequency of less than three (3).*
- *Test the learnt classifier using enronemail_1h.txt and report the misclassification error rate. Remember to use all white-space delimitted tokens as independent input variables (assume spaces, fullstops, commas as delimiters). How do we treat tokens in the test set that do not appear in the training set?*

HW2.8.1 OPTIONAL

- *Run both the Multinomial Naive Bayes and the Bernoulli Naive Bayes algorithms from SciKit-Learn (using default settings) over the same training data used in HW2.8 and report the misclassification error on both the training set and the testing set*
- *Prepare a table to present your results, where rows correspond to approach used (SciKit-Learn Multinomial NB; SciKit-Learn Bernouili NB; Your Hadoop implementation) and the columns presents the training misclassification error, and the misclassification error on the test data set*
- *Discuss the performance differences in terms of misclassification error rates over the test and training datasets by the different implementations. Which approch (Bernouili versus Multinomial) would you recommend for SPAM detection? Justify your selection.*

In [ ]: