MIDS-W261-HW-09-TEMPLATE

November 15, 2016

```
In [ ]: %%javascript
        /*********************************************************************************************
        Known Mathjax Issue with Chrome - a rounding issue adds a border to the right of mathjax markup
        https://github.com/mathjax/MathJax/issues/1300
        A quick hack to fix this based on stackoverflow discussions:
        http://stackoverflow.com/questions/34277967/chrome-rendering-mathjax-equations-with-a-trailing-
        *********************************************************************************************/

        $('.math>span').css("border-left-color","transparent")
```

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

1 MIDS - w261 Machine Learning At Scale

Course Lead: Dr James G. Shanahan (**email** Jimi via James.Shanahan AT gmail.com)

1.1 Assignment - HW9

Name: Anthony Spalvieri-Kruse
Class: MIDS w261 (Section 1, e.g., Fall 2016 Group 1)
Email: ask@iSchool.Berkeley.edu
Unit 9:
Due Time: HW9 is due on FRIDAY 11/11/2016.

2 Table of Contents

1 Instructions Back to Table of Contents

MIDS UC Berkeley, Machine Learning at Scale DATSCIW261 ASSIGNMENT #9
Version 2016-11-01

2.0.1 INSTRUCTIONS for SUBMISSIONS

Please use the following form for HW submission:
https://docs.google.com/forms/d/1ZOr9RnIe_A06AcZDB6K1mJN4vrLeSmS2PD6Xm3eOiis/viewform?usp=send_form

2.0.2 IMPORTANT

HW9 can be completed locally on your computer for most part but will require a cluster of computers for the bigger wikipedia dataset.

2.0.3 Documents:

- IPython Notebook, published and viewable online.
- PDF export of IPython Notebook.

2 Useful References Back to Table of Contents

- See async and live lectures for this week
- Data-intensive text processing with MapReduce. San Rafael, CA: Morgan & Claypool Publishers. Chapter 5.

HW Problems Back to Table of Contents

HW 9 Dataset
Note that all referenced files life in the enclosing directory. Checkout the Data subdirectory on Dropbox or the AWS S3 buckets (details contained each question).

2.1 3. HW9.0 Short answer questions

Back to Table of Contents

__ What is PageRank and what is it used for in the context of web search?__

```
Page rank is a graph algorithm that performs iterative matrix multiplications on an arbitrary starting
```

__ What modifications have to be made to the webgraph in order to leverage the machinery of Markov Chains to compute the Steady State Distibution? __

```
We need to add teleportation and a damping factor.  Teleportation is to allow dangling nodes to pass th
```

__ OPTIONAL: In topic-specific pagerank, how can we ensure that the irreducible property is satifsied? (HINT: see HW9.4) __

```
We would have to just remove the nodes that can't be reached by in-topic nodes.
```

2.2 3. HW9.1 MRJob implementation of basic PageRank

Back to Table of Contents

Write a basic MRJob implementation of the iterative PageRank algorithm that takes sparse adjacency lists as input (as explored in HW 7).

Make sure that you implementation utilizes teleportation (1-damping/the number of nodes in the network), and further, distributes the mass of dangling nodes with each iteration so that the output of each iteration is correctly normalized (sums to 1).

[NOTE: The PageRank algorithm assumes that a random surfer (walker), starting from a random web page, chooses the next page to which it will move by clicking at random, with probability d,one of the hyperlinks in the current page. This probability is represented by a so-called damping factor d, where d $\in$ (0, 1). Otherwise, with probability (1 − d), the surfer jumps to any web page in the network. If a page is a dangling end, meaning it has no outgoing hyperlinks, the random surfer selects an arbitrary web page from a uniform distribution and "teleports" to that page]

As you build your code, use the test data:

s3://ucb-mids-mls-networks/PageRank-test.txt

Or under the Data Subfolder for HW7 on Dropbox with the same file name. > Dropbox: https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAAKsjQfF9uHfv-X9mCqr9wa?dl=0

with teleportation parameter set to 0.15 (1-d, where d, the damping factor is set to 0.85), and crosscheck your work with the true result, displayed in the first image in the Wikipedia article and here for reference are the corresponding PageRank probabilities:

```
In [7]: %%writefile prEDA.py
        #!/usr/bin/env python

        from mrjob.job import MRJob
        from mrjob.step import MRStep
        import ast
        import itertools

        class EDA(MRJob):

            def configure_options(self):
                super(EDA, self).configure_options()
                self.add_passthrough_option('--feature', type = "str")

            def __init__(self, *args, **kwargs):
                super(EDA, self).__init__(*args, **kwargs)
                self.feature = self.options.feature
                #Getting specific nodes fails for big data
                self.nodes = set()
                self.frequency = 0
                self.numLinks = 0
                self.minDegree = 0
                self.maxDegree = 0
                self.options.jobconf = {"mapred.reduce.tasks":1 }

            def mapper(self, key, line):
                node, neighbors = line.strip().split("\t")
                neighbors = ast.literal_eval(neighbors)
                numLinks = len(neighbors)
                #self.nodes.add(node)
                #for neighbor in neighbors:
```

```
        #    self.nodes.add(neighbor)

        if self.feature == "Nodes" or self.feature == "numNodes":
            for neighbor in neighbors:
                yield neighbor, 1
            yield node, 1
        elif self.feature == "numLinks":
            yield node, numLinks
        elif self.feature == "averageDegree":
            yield node, numLinks
        elif self.feature == "minMaxDegree":
            yield None, numLinks
        else:
            print "Acceptable features: numNodes, numLinks, averageDegree, minMaxDegree"
            exit()

    def combiner(self, key, values):
        if self.feature == "Nodes" or self.feature == "numNodes":
            yield key, 1
        elif self.feature == "numLinks":
            yield key, sum(values)
        elif self.feature == "averageDegree":
            yield key, sum(values)
        elif self.feature == "minMaxDegree":
            values, valuesCp = itertools.tee(values)
            yield None, (min(values), max(valuesCp))

    def reducer(self, key, values):
        if self.feature == "Nodes" or self.feature == "numNodes":
            self.frequency+=1
            #self.nodes.add(key)
        elif self.feature == "numLinks":
            self.numLinks += sum(values)
        elif self.feature == "averageDegree":
            self.frequency+=1
            self.numLinks += sum(values)
        elif self.feature == "minMaxDegree":
            values, valuesCp = itertools.tee(values)
            self.maxDegree = max(values)
            self.minDegree = min(valuesCp)

    def reducer_final(self):
        if self.feature == "numNodes":
            yield "Number of Nodes: ", self.frequency
        elif self.feature == "numLinks":
            yield "Number of Links", self.numLinks
        elif self.feature == "averageDegree":
            yield "Average Degree: ", float(self.numLinks)/self.frequency
        elif self.feature == "minMaxDegree":
            yield "Min/Max Degree: ", (self.minDegree, self.maxDegree)
        elif self.feature == "Nodes":
            yield "", [node for node in self.nodes]
```

```
            def steps(self):
                return [MRStep(mapper=self.mapper,reducer=self.reducer,
                               reducer_final=self.reducer_final)]

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

Overwriting prEDA.py
```

```
In [162]: !./prEDA.py PageRank-test.txt --feature "numNodes"
          !./prEDA.py PageRank-test.txt --feature "Nodes" --output "testNodes"
          !./prEDA.py PageRank-test.txt --feature "numLinks"
          !./prEDA.py PageRank-test.txt --feature "averageDegree"
          !./prEDA.py PageRank-test.txt --feature "minMaxDegree"

No configs found; falling back on auto-configuration
Creating temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
Running step 1 of 1...
Streaming final output from /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKrus
"Number of Nodes: "        11
Removing temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
No configs found; falling back on auto-configuration
Running step 1 of 1...
Creating temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
Streaming final output from testNodes.txt...
""         ["A", "C", "B", "E", "D", "G", "F", "I", "H", "K", "J"]
Removing temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
No configs found; falling back on auto-configuration
Creating temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
Running step 1 of 1...
Streaming final output from /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKrus
"Number of Links"        17
Removing temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
No configs found; falling back on auto-configuration
Creating temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
Running step 1 of 1...
Streaming final output from /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKrus
"Average Degree: "        1.7
Removing temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
No configs found; falling back on auto-configuration
Creating temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
Running step 1 of 1...
Streaming final output from /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKrus
"Min/Max Degree: "        [1, 3]
Removing temp directory /var/folders/tl/kz2xr4bd3dvf07wcdjmvpl680000gn/T/prEDA.AnthonySpalvieriKruse.20
```

HW 9.1 Implementation

```
In [1]: %%writefile PageRankAlternate2.py
        #!/usr/bin/env python

        #This was my second attempt to rewrite my job to work at scale, it also failed.

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

```
from collections import defaultdict
import ast

class PageRankAlternate2(MRJob):

    def configure_options(self):
        super(PageRankAlternate2, self).configure_options()
        self.add_passthrough_option('--iterations', type = "int", default=1)
        self.add_passthrough_option('--N', type = "int", default = -1)
        self.add_passthrough_option('--alpha', type = "float", default = .15)

    def __init__(self, *args, **kwargs):
        super(PageRankAlternate2, self).__init__(*args, **kwargs)
        self.iterations = self.options.iterations
        self.N = self.options.N
        self.alpha = self.options.alpha
        self.loss = 0
        self.inflator = 10**10
        self.options.jobconf = {"mapred.reduce.tasks":1 }


    def initialize_graph_map(self, _, line):
        node, links = line.strip().split("\t")
        links = ast.literal_eval(links)
        for link in links:
            yield link, {}
        yield node, links

    def initialize_graph_combine(self, node, lines):
        links = {}
        for item in lines:
            if len(item)!=0:
                links = item
        yield node, links

    def initialize_graph_reduce(self, node, lines):
        links = {}
        for item in lines:
            if len(item)!=0:
                links = item
        yield node, (links, float(1)/self.N)

    #Because lines are unique, each node here will occur only once
    #If it's got links, it's not dangling, and we have 0 for dangling cell
    #if it lacks links, it is dangling, so we add it's nodeProb to dangling
    def pass_mass(self, node, line):
        links, prob = line
        #Gets called on the first iteration
        if isinstance(links, basestring):
            links = ast.literal_eval(links)

        numLinks = len(links)
        if numLinks == 0:
            self.loss += prob * self.inflator
```

```
            #Format: yield node, (links, passedMass, danglingLossMass)
            yield node, (links, 0)
        else:
            for link in links:
                yield link, ({}, float(prob)/numLinks)
            yield node, (links, 0)

    def mapper_final(self):
        yield "0", (self.loss)

    def combiner(self, node, lines):
        if node == "0":
            loss = sum(lines)
            yield node, loss
        else:
            nodeProb = 0
            links = {}
            for item in lines:
                link, passedMass = item
                nodeProb+=passedMass
                if len(link)!=0:
                    links = link
            yield node, (links, nodeProb)

    def sum_mass(self, node, lines):
        if node == "0":
            self.loss = sum(lines)
        else:
            loss = float(self.loss)/self.inflator
            total=0
            links = {}
            G = self.N
            a = self.alpha
            for item in lines:
                link, prob = item
                total+=prob
                if len(link) != 0:
                    links = link

            updatedProb = float(a)/G + (1-a)*(float(loss)/G + total)
            yield node, (links, updatedProb)

    def steps(self):
        return (
                [MRStep(mapper=self.initialize_graph_map,
                        combiner = self.initialize_graph_combine,
                        reducer = self.initialize_graph_reduce)] +
                [MRStep(mapper = self.pass_mass, mapper_final = self.mapper_final,
                        combiner=self.combiner,
                        reducer = self.sum_mass)] * self.options.iterations
        )

if __name__ == '__main__':
    PageRankAlternate2.run()
```

```
Writing PageRankAlternate2.py
```

```
In [5]: %%writefile PageRankAlternate.py
        #!/usr/bin/env python


        #This was my first attempt to rewrite my job to work at scale, it failed.

        from mrjob.job import MRJob
        from mrjob.step import MRStep
        from collections import defaultdict
        import ast

        class PageRankAlternate(MRJob):

            def configure_options(self):
                super(PageRankAlternate, self).configure_options()
                self.add_passthrough_option('--iterations', type = "int", default=1)
                self.add_passthrough_option('--N', type = "int", default = -1)
                self.add_passthrough_option('--alpha', type = "float", default = .15)

            def __init__(self, *args, **kwargs):
                super(PageRankAlternate, self).__init__(*args, **kwargs)
                self.iterations = self.options.iterations
                self.N = self.options.N
                self.alpha = self.options.alpha

            def initialize_graph_map(self, _, line):
                node, links = line.strip().split("\t")
                links = ast.literal_eval(links)
                for link in links:
                    yield link, {}
                yield node, links

            def initialize_graph_combine(self, node, lines):
                links = {}
                for item in lines:
                    if len(item)!=0:
                        links = item
                yield node, links

            def initialize_graph_reduce(self, node, lines):
                links = {}
                for item in lines:
                    if len(item)!=0:
                        links = item
                yield node, (links, float(1)/self.N)

            #Because lines are unique, each node here will occur only once
            #If it's got links, it's not dangling, and we have 0 for dangling cell
            #if it lacks links, it is dangling, so we add it's nodeProb to dangling
            def pass_mass(self, node, line):
                links, prob = line
                #Gets called on the first iteration
```

```
        if isinstance(links, basestring):
            links = ast.literal_eval(links)

        numLinks = len(links)
        if numLinks == 0:
            #Format: yield node, (links, passedMass, danglingLossMass)
            yield node, (links, 0, prob)
        else:
            for link in links:
                yield link, ({}, float(prob)/numLinks, 0)
            yield node, (links, 0, 0)

    def combiner(self, node, lines):
        nodeProb = 0
        partialLossMass = 0
        links = {}
        for item in lines:
            link, passedMass, lossMass = item
            nodeProb+=passedMass
            partialLossMass += lossMass
            if len(link)!=0:
                links = link
        yield None, (node, links, nodeProb, partialLossMass)

    def sum_mass(self, _, lines):
        loss = 0
        links = {}
        G = self.N
        a = self.alpha
        nodesLinks = defaultdict(dict)
        nodesProbs = defaultdict(int)
        for line in lines:
            node, links, nodeProb, partialLossMass = line
            loss+=partialLossMass
            nodesProbs[node]+=nodeProb
            #ensures all nodes get a links dict
            if node not in nodesLinks:
                nodesLinks[node] = links
            #there's only one non-empty links dict per node, so we only replace one or fewer ti
            if len(links) != 0:
                nodesLinks[node] = links


        for nod in nodesLinks.keys():
            updatedProb = float(a)/G + (1-a)*(float(loss)/G + nodesProbs[nod])
            yield nod, (nodesLinks[nod], updatedProb)

    def steps(self):
        return (
                [MRStep(mapper=self.initialize_graph_map,
                        combiner = self.initialize_graph_combine,
                        reducer = self.initialize_graph_reduce)] +
                [MRStep(mapper = self.pass_mass,
                        combiner=self.combiner,
```

```
                                reducer = self.sum_mass)] * self.options.iterations
            )

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

```
Writing PageRankAlternate.py
```

In [3]:
```
%%writefile PageRank.py
#!/usr/bin/env python
##!/Users/AnthonySpalvieriKruse/anaconda/bin/python

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

class PageRank(MRJob):

    def configure_options(self):
        super(PageRank, self).configure_options()
        self.add_passthrough_option('--iterations', type = "int", default=1)
        self.add_passthrough_option('--N', type = "int", default = -1)
        self.add_passthrough_option('--alpha', type = "float", default = .15)

    def __init__(self, *args, **kwargs):
        super(PageRank, self).__init__(*args, **kwargs)
        self.iterations = self.options.iterations
        self.N = self.options.N
        self.alpha = self.options.alpha

    def initialize_graph(self, _, line):
        node, links = line.strip().split("\t")
        yield node, (links, float(1)/self.N)

    def pass_mass(self, node, line):
        links, prob = line

        #Gets called on the first iteration
        if isinstance(links, basestring):
            links = ast.literal_eval(links)

        numLinks = len(links)

        #
        if numLinks == 0:
            yield "dangling", prob
        else:
            for link in links:
                yield link, prob/numLinks

        yield node, links

    def combiner(self, node, lines):
        if node == "dangling":
            yield node, sum(lines)
```

```
                else:
                    total = 0
                    links = {}
                    for item in lines:
                        if type(item) is float:
                            total+=item
                        else:
                            links = item
                            yield node, links
                    yield node, total

            def sum_mass(self, node, lines):
                if node == "dangling":
                    loss = sum(lines)
                    #This assumes the nodes are indexed, which could fail
                    #Better idea is to pass file w/ all nodes, and iterate through this list
                    for i in range(1, int(self.options.N) + 1):
                        yield str(i), loss
                else:
                    total = 0
                    links = {}
                    for item in lines:
                        if type(item) is float:
                            total+=item
                        else:
                            links = item
                    yield node, (links, total)

            def teleport(self, node, lines):
                G = self.N
                a = self.alpha
                loss = 0

                for item in lines:
                    if type(item) == float:
                        loss = item
                    else:
                        links, prob = item

                updatedProb = float(a)/G + (1-a)*(float(loss)/G + prob)
                yield node, (links, updatedProb)

            def steps(self):
                return (
                        [MRStep(mapper=self.initialize_graph)] +
                        [MRStep(mapper = self.pass_mass, combiner = self.combiner, reducer = self.sum_ma
                         MRStep(reducer = self.teleport)] * self.options.iterations
                )

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

```
Overwriting PageRank.py
```

In [ ]:

In [1]:
```
%%writefile PageRankPreprocess.py
#!/usr/bin/env python

#This is the first job i use in my scaled version of this job

from mrjob.job import MRJob
from mrjob.step import MRStep
from collections import defaultdict
import ast

class PageRankPreprocess(MRJob):

    def configure_options(self):
        super(PageRankPreprocess, self).configure_options()
        self.add_passthrough_option('--N', type = "int", default = -1)

    def __init__(self, *args, **kwargs):
        super(PageRankPreprocess, self).__init__(*args, **kwargs)
        self.N = self.options.N
        self.inflator = 10**12

    def initialize_graph_map(self, _, line):
        node, links = line.strip().split("\t")
        links = ast.literal_eval(links)
        for link in links:
            yield link, {}
        yield node, links

    def initialize_graph_combine(self, node, lines):
        links = {}
        for item in lines:
            if len(item)!=0:
                links = item
        yield node, links

    def initialize_graph_reduce(self, node, lines):
        links = {}
        for item in lines:
            if len(item)!=0:
                links = item
        #if dangling increment loss counter
        if len(links) == 0:
            self.increment_counter('loss', 'lossCounter', int(self.inflator*float(1)/self.N))

        yield node, (links, float(1)/self.N)

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

if __name__ == '__main__':
```

```
            PageRankPreprocess.run()

Writing PageRankPreprocess.py
```

In [2]:
```
%%writefile PageRankScale.py
#!/usr/bin/env python

from mrjob.job import MRJob
from mrjob.step import MRStep
from collections import defaultdict
import ast

class PageRankScale(MRJob):

    def configure_options(self):
        super(PageRankScale, self).configure_options()
        self.add_passthrough_option('--N', type = "int", default = -1)
        self.add_passthrough_option('--alpha', type = "float", default = .15)
        self.add_passthrough_option('--loss', type = "float", default = 0)

    def __init__(self, *args, **kwargs):
        super(PageRankScale, self).__init__(*args, **kwargs)
        self.N = self.options.N
        self.alpha = self.options.alpha
        self.inflator = 10**12
        self.loss = float(self.options.loss)/self.inflator

    def pass_mass(self, node, line):
        node, linksAndProb = map(lambda x: ast.literal_eval(x), line.split('\t'))
        links, prob = linksAndProb
        if isinstance(links, basestring):
            links = ast.literal_eval(links)

        numLinks = len(links)
        for link in links:
            yield link, ({}, float(prob)/numLinks)
        yield node, (links, 0)


    def combiner(self, node, lines):
        nodeProb = 0
        links = {}
        for item in lines:
            link, passedMass = item
            nodeProb+=passedMass
            if len(link)!=0:
                links = link
        yield node, (links, nodeProb)

    def sum_mass(self, node, lines):
        total=0
        links = {}
        G = self.N
        a = self.alpha
        for item in lines:
```

```
                link, prob = item
                total+=prob
                if len(link) != 0:
                    links = link

            updatedProb = float(a)/G + (1-a)*(float(self.loss)/G + total)

            if len(links) == 0:
                self.increment_counter('loss', 'lossCounter', int(self.inflator*updatedProb))

            yield node, (links, updatedProb)

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

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

```
Writing PageRankScale.py
```

```
In [1]: %%writefile PageRankRunnerScale.py
        #!/usr/bin/env python

        from numpy import random,array
        from PageRankScale import PageRankScale
        from PageRankPreprocess import PageRankPreprocess
        import sys


        numIterations = sys.argv[2]
        mr_job = PageRankPreprocess(args=[sys.argv[1], "-r", "hadoop", "--output-dir", "preprocessed0/"
        i=0
        print numIterations
        for i in range(0,int(numIterations)):
            print "iteration: ", i

            with mr_job.make_runner() as runner:
                runner.run()
                lossMap = runner.counters()
                loss = lossMap[0]["loss"]["lossCounter"]
                print loss

            mr_job = PageRankScale(args=['hdfs:/user/ask/preprocessed'+str(i)+'/', "-r", "hadoop", "--N

            i += 1
```

```
Writing PageRankRunnerScale.py
```

3 9.1 Solution

In [15]:
```
#9.1 actual solution here
!./PageRankAlternate.py PageRank-test_indexed.txt --iterations 15 --N 11 --alpha .15
```

```
No configs found; falling back on auto-configuration
Creating temp directory /tmp/PageRankAlternate.ask.20161116.002142.868830
Running step 1 of 16...
Running step 2 of 16...
Running step 3 of 16...
Running step 4 of 16...
Running step 5 of 16...
Running step 6 of 16...
Running step 7 of 16...
Running step 8 of 16...
Running step 9 of 16...
Running step 10 of 16...
Running step 11 of 16...
Running step 12 of 16...
Running step 13 of 16...
Running step 14 of 16...
Running step 15 of 16...
Running step 16 of 16...
Streaming final output from /tmp/PageRankAlternate.ask.20161116.002142.868830/output...
"11"        [{"5": 1}, 0.01617057945713026]
"10"        [{"5": 1}, 0.01617057945713026]
"1"        [{}, 0.032792273760929885]
"3"        [{"2": 1}, 0.33367485020657883]
"2"        [{"3": 1}, 0.3935841617821781]
"5"        [{"2": 1, "4": 1, "6": 1}, 0.08090526589921213]
"4"        [{"1": 1, "2": 1}, 0.03909527553272476]
"7"        [{"2": 1, "5": 1}, 0.01617057945713026]
"6"        [{"2": 1, "5": 1}, 0.03909527553272476]
"9"        [{"2": 1, "5": 1}, 0.01617057945713026]
"8"        [{"2": 1, "5": 1}, 0.01617057945713026]
Removing temp directory /tmp/PageRankAlternate.ask.20161116.002142.868830...
```

In [2]:
```
%%writefile PageRankRunner.py
#!/usr/bin/env python
##!/Users/AnthonySpalvieriKruse/anaconda/bin/python

from numpy import random,array
from PageRank import PageRank
import sys

def runPageRank(alpha, output):
    mr_job = PageRank(args=["PageRank-test_indexed.txt", "-r", "hadoop", "--iterations", "12",
                            "11", "--alpha", alpha, "--output", output])
    visited = set()
    firstPass = True

    with mr_job.make_runner() as runner:
            runner.run()

    # stream_output: get access of the output
```

```
            for line in runner.stream_output():
                # value is the gradient value
                node, value =  mr_job.parse_output_line(line)
                print node, value

        if __name__ == "__main__":
            runPageRank(sys.argv[1], sys.argv[2])
```

```
Overwriting PageRankRunner.py
```

HW 9.1 Analysis

3.1 3. HW9.2: Exploring PageRank teleportation and network plots

Back to Table of Contents

- In order to overcome problems such as disconnected components, the damping factor (a typical value for d is 0.85) can be varied.
- Using the graph in HW1, plot the test graph (using networkx, https://networkx.github.io/) for several values of the damping parameter alpha, so that each nodes radius is proportional to its PageRank score.
- In particular you should do this for the following damping factors: [0,0.25,0.5,0.75, 0.85, 1].
- Note your plots should look like the following: https://en.wikipedia.org/wiki/PageRank#/media/File:PageRanks-Example.svg

HW 9.2 Implementation

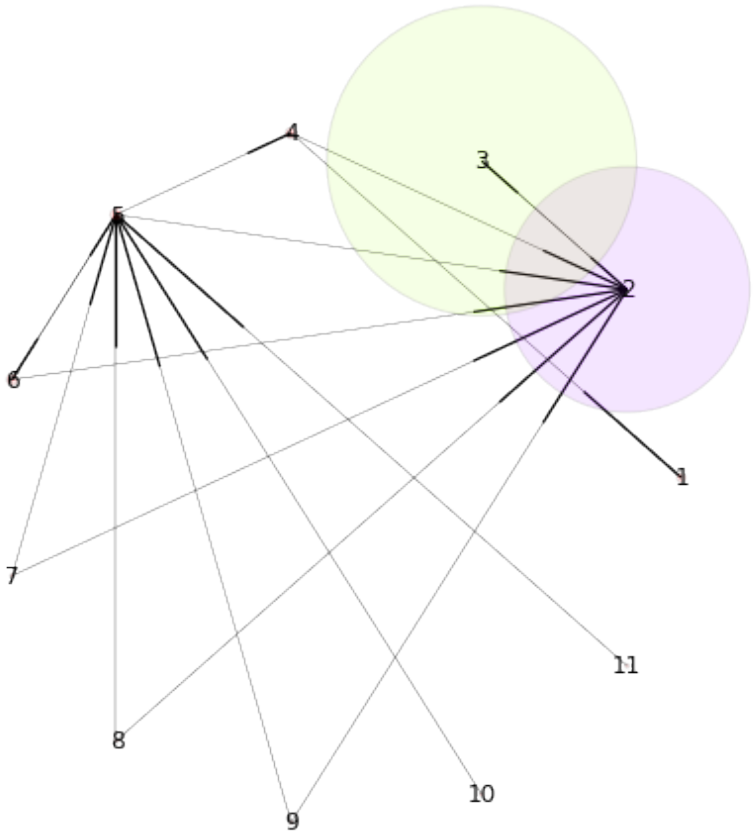
In [128]:
```
import PageRankRunner as prr

alphas = [0,0.25,0.5,0.75, 0.85, 1]

for alpha in alphas:
    prr.runPageRank(alpha, "output/graph_{}".format(alpha))
```

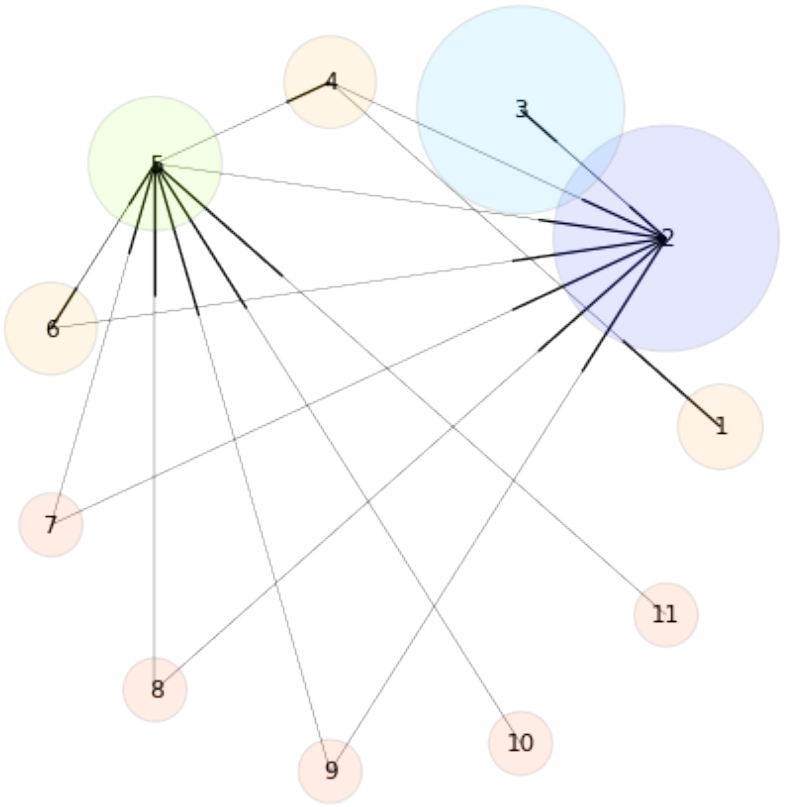
```
1 [{}, 0.000520591833482673]
10 [{u'5': 1}, 0.00010558523000447849]
11 [{u'5': 1}, 0.00010558523000447849]
2 [{u'3': 1}, 0.38400157742994956]
3 [{u'2': 1}, 0.6124293358239826]
4 [{u'1': 1, u'2': 1}, 0.0008050824482794051]
5 [{u'2': 1, u'4': 1, u'6': 1}, 0.0009104038660036321]
6 [{u'2': 1, u'5': 1}, 0.0008050824482794051]
7 [{u'2': 1, u'5': 1}, 0.00010558523000447849]
8 [{u'2': 1, u'5': 1}, 0.00010558523000447849]
9 [{u'2': 1, u'5': 1}, 0.00010558523000447849]
1 [{}, 0.04630914414862571]
10 [{u'5': 1}, 0.02588609844107977]
11 [{u'5': 1}, 0.02588609844107977]
2 [{u'3': 1}, 0.32558170861680436]
3 [{u'2': 1}, 0.27549669005271776]
4 [{u'1': 1, u'2': 1}, 0.05446066550463443]
5 [{u'2': 1, u'4': 1, u'6': 1}, 0.11426063396718444]
6 [{u'2': 1, u'5': 1}, 0.05446066550463443]
7 [{u'2': 1, u'5': 1}, 0.02588609844107977]
8 [{u'2': 1, u'5': 1}, 0.02588609844107977]
9 [{u'2': 1, u'5': 1}, 0.02588609844107977]
```

```
1 [{}, 0.06694783807417107]
10 [{u'5': 1}, 0.04849763611502909]
11 [{u'5': 1}, 0.04849763611502909]
2 [{u'3': 1}, 0.22841211108509352]
3 [{u'2': 1}, 0.16273156301660469]
4 [{u'1': 1, u'2': 1}, 0.0738008017499569]
5 [{u'2': 1, u'4': 1, u'6': 1}, 0.15181870374907153]
6 [{u'2': 1, u'5': 1}, 0.0738008017499569]
7 [{u'2': 1, u'5': 1}, 0.04849763611502909]
8 [{u'2': 1, u'5': 1}, 0.04849763611502909]
9 [{u'2': 1, u'5': 1}, 0.04849763611502909]
1 [{}, 0.0802296662490276]
10 [{u'5': 1}, 0.07000521968834601]
11 [{u'5': 1}, 0.07000521968834601]
2 [{u'3': 1}, 0.15573090701925202]
3 [{u'2': 1}, 0.10893794984542454]
4 [{u'1': 1, u'2': 1}, 0.08179557248396667]
5 [{u'2': 1, u'4': 1, u'6': 1}, 0.1414842334766323]
6 [{u'2': 1, u'5': 1}, 0.08179557248396667]
7 [{u'2': 1, u'5': 1}, 0.07000521968834601]
8 [{u'2': 1, u'5': 1}, 0.07000521968834601]
9 [{u'2': 1, u'5': 1}, 0.07000521968834601]
1 [{}, 0.08478337347441553]
10 [{u'5': 1}, 0.07842886418374316]
11 [{u'5': 1}, 0.07842886418374316]
2 [{u'3': 1}, 0.1297663766636617]
3 [{u'2': 1}, 0.09789382068773601]
4 [{u'1': 1, u'2': 1}, 0.08472679054229493]
5 [{u'2': 1, u'4': 1, u'6': 1}, 0.12595852717088107]
6 [{u'2': 1, u'5': 1}, 0.08472679054229493]
7 [{u'2': 1, u'5': 1}, 0.07842886418374316]
8 [{u'2': 1, u'5': 1}, 0.07842886418374316]
9 [{u'2': 1, u'5': 1}, 0.07842886418374316]
1 [{}, 0.09090909090909091]
10 [{u'5': 1}, 0.09090909090909091]
11 [{u'5': 1}, 0.09090909090909091]
2 [{u'3': 1}, 0.09090909090909091]
3 [{u'2': 1}, 0.09090909090909091]
4 [{u'1': 1, u'2': 1}, 0.09090909090909091]
5 [{u'2': 1, u'4': 1, u'6': 1}, 0.09090909090909091]
6 [{u'2': 1, u'5': 1}, 0.09090909090909091]
7 [{u'2': 1, u'5': 1}, 0.09090909090909091]
8 [{u'2': 1, u'5': 1}, 0.09090909090909091]
9 [{u'2': 1, u'5': 1}, 0.09090909090909091]
```

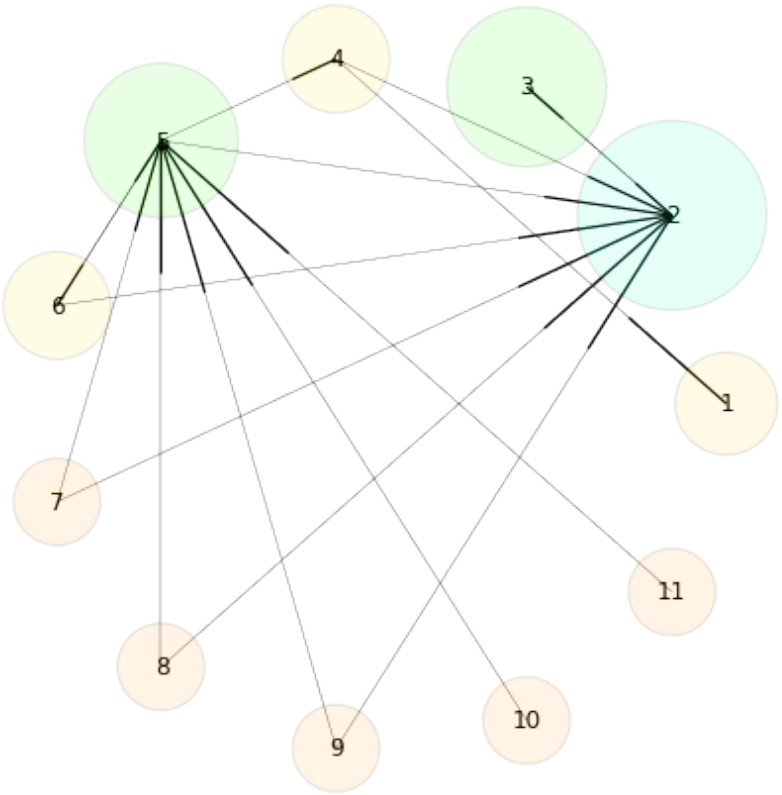
In [209]:
```
%matplotlib inline

import networkx as nx
import ast
from matplotlib import pyplot as plt
import os
import colorsys
```

```
def draw_graph(path):
    plt.figure(figsize=(8, 8))
    probs = {}
    graph = nx.DiGraph()

    for fname in os.listdir(path):
        with open(path+"/"+fname) as f:
            for line in f:
                node, linksAndProb = line.strip().split("\t")
                node = node.replace('"','')
                links, prob = ast.literal_eval(linksAndProb)
                probs[int(node)] = prob

                for link in links:
                    graph.add_edge(int(node), int(link))

    weights = [probs[n]*40000 for n in graph.nodes()]
    layout = nx.circular_layout(graph)

    # set labels and colors
    labels = {n:n for n in graph.nodes()}

    HSV_tuples = [(x*2, 1, 1) for x in probs.values()]
    node_colors = map(lambda x: colorsys.hsv_to_rgb(*x), HSV_tuples)

    # draw graph
    nx.draw_networkx_nodes(graph, layout, node_size = weights, alpha = 0.1, node_color = node
    nx.draw_networkx_edges(graph, layout, width = 0.3, alpha = 1, edge_color = 'black')
    nx.draw_networkx_labels(graph, layout, labels=labels, font_size = 12)
    # show graph
    plt.title("alpha={}".format(path))
    plt.axis('off')
    plt.tight_layout()
    plt.show()


for path in os.listdir("output"):
    draw_graph("output/"+path)
```

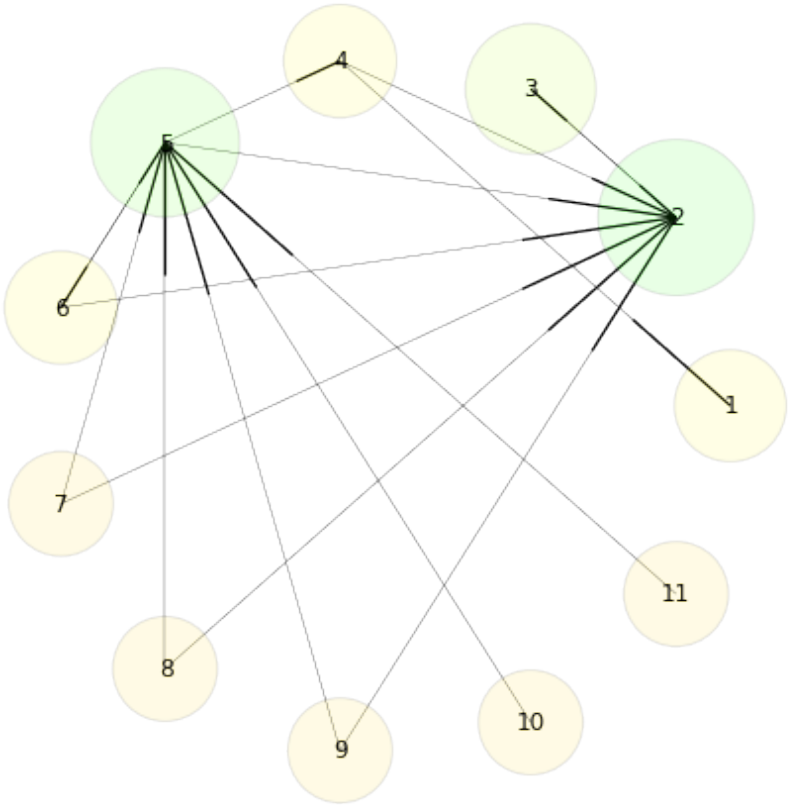

alpha=output/graph_0
4
3
5
2
6
1
7
11
8
10
9

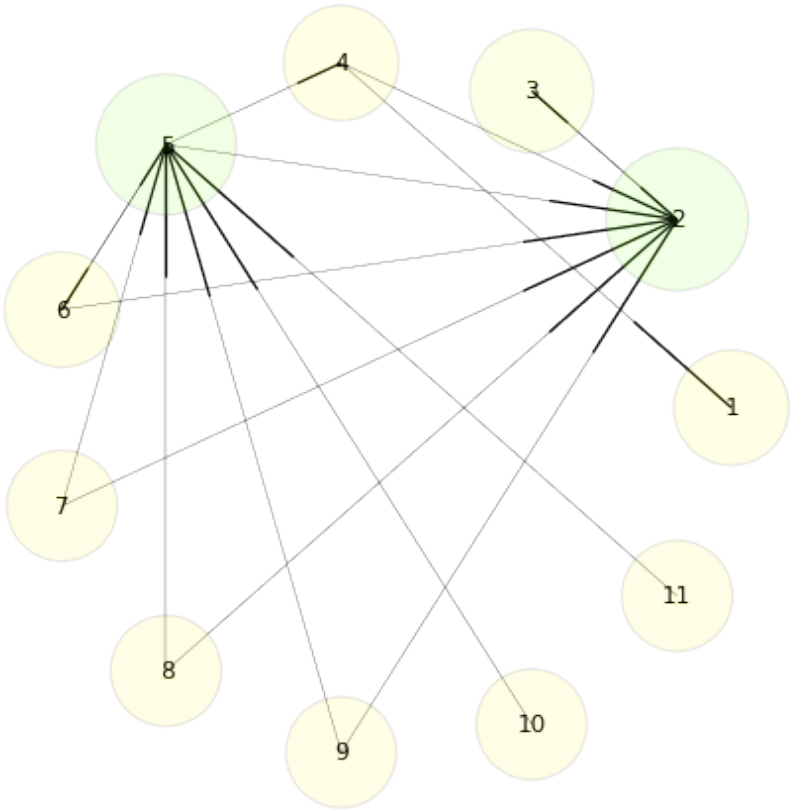

alpha=output/graph_0.25
4
3
5
2
6
1
7
11
8
10
9

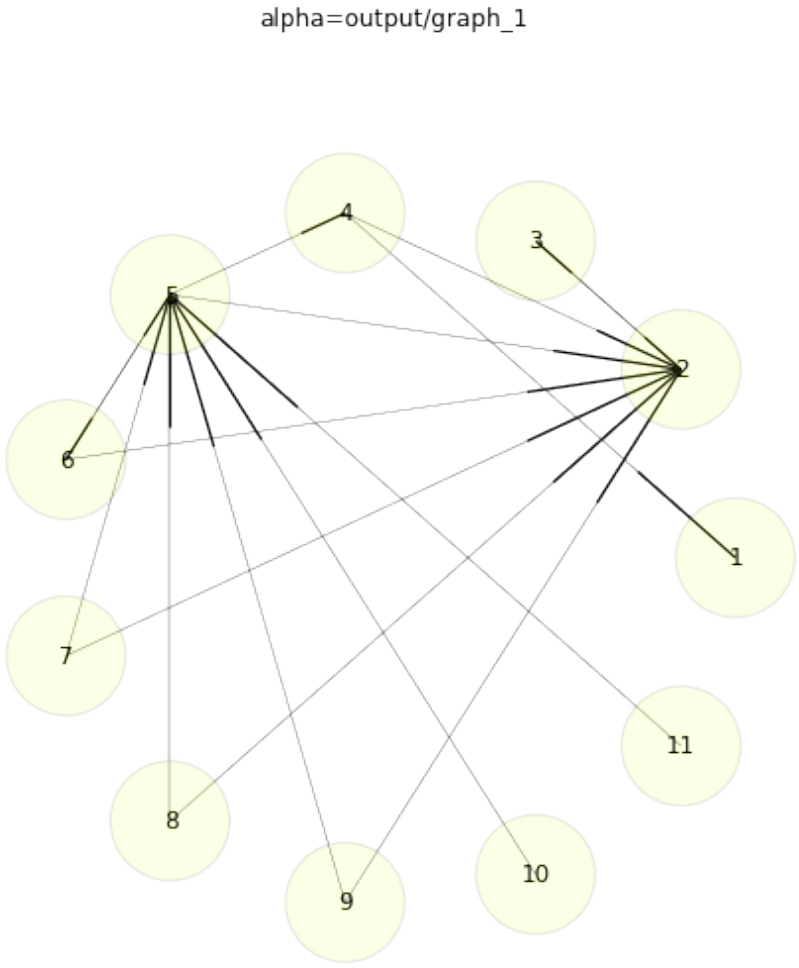
alpha=output/graph_0.5



alpha=output/graph_0.75




alpha=output/graph_0.85
4
3
5
2
6
1
7
11
8
10
9



HW 9.2 Analysis

3.2 3. HW9.3: Applying PageRank to the Wikipedia hyperlinks network

Back to Table of Contents

< * Run your PageRank implementation on the Wikipedia dataset for 5 iterations, and display the top 100 ranked nodes (with alpha = 0.85). * Run your PageRank implementation on the Wikipedia dataset for 10 iterations, and display the top 100 ranked nodes (with teleportation factor of 0.15). * Have the top 100 ranked pages changed? Comment on your findings. * Plot the pagerank values for the top 100 pages resulting from the 5 iterations run. Then plot the pagerank values for the same 100 pages that resulted from the 10 iterations run.

HW 9.3 Implementation

```
In [8]: #Get the total number of nodes
        !./prEDA.py all-pages-indexed-out.txt -r hadoop --feature "numNodes"
```

```
No configs found; falling back on auto-configuration
Creating temp directory /tmp/prEDA.ask.20161112.174047.514269
Looking for hadoop binary in /opt/hadoop/bin...
Found hadoop binary: /opt/hadoop/bin/hadoop
Using Hadoop version 2.7.2
Copying local files to hdfs:///user/ask/tmp/mrjob/prEDA.ask.20161112.174047.514269/files/...
Looking for Hadoop streaming jar in /opt/hadoop...
Found Hadoop streaming jar: /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar
Detected hadoop configuration property names that do not match hadoop version 2.7.2:
The have been translated as follows
 mapred.reduce.tasks: mapreduce.job.reduces
Running step 1 of 1...
  packageJobJar: [] [/opt/hadoop-2.7.2/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar] /tmp/streamjo
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Loaded native gpl library from the embedded binaries
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev d62701d4d05dfa6115bbaf8d9dff002d
  Total input paths to process : 1
  number of splits:8
  Submitting tokens for job: job_1475792131280_1207
  Submitted application application_1475792131280_1207
  The url to track the job: http://rm-ia.s3s.altiscale.com:8088/proxy/application_1475792131280_1207/
  Running job: job_1475792131280_1207
  Job job_1475792131280_1207 running in uber mode : false
   map 0% reduce 0%
   map 1% reduce 0%
   map 2% reduce 0%
   map 3% reduce 0%
   map 4% reduce 0%
   map 5% reduce 0%
   map 6% reduce 0%
   map 7% reduce 0%
   map 8% reduce 0%
   map 9% reduce 0%
   map 10% reduce 0%
   map 11% reduce 0%
   map 12% reduce 0%
   map 13% reduce 0%
   map 14% reduce 0%
   map 15% reduce 0%
   map 16% reduce 0%
   map 17% reduce 0%
   map 18% reduce 0%
   map 19% reduce 0%
   map 20% reduce 0%
   map 21% reduce 0%
   map 22% reduce 0%
   map 23% reduce 0%
   map 24% reduce 0%
   map 25% reduce 0%
```

```
map 26% reduce 0%
map 27% reduce 0%
map 28% reduce 0%
map 29% reduce 0%
map 30% reduce 0%
map 31% reduce 0%
map 32% reduce 0%
map 33% reduce 0%
map 34% reduce 0%
map 35% reduce 0%
map 36% reduce 0%
map 37% reduce 0%
map 38% reduce 0%
map 39% reduce 0%
map 40% reduce 0%
map 41% reduce 0%
map 42% reduce 0%
map 43% reduce 0%
map 44% reduce 0%
map 45% reduce 0%
map 46% reduce 0%
map 47% reduce 0%
map 48% reduce 0%
map 49% reduce 0%
map 50% reduce 0%
map 51% reduce 0%
map 52% reduce 0%
map 53% reduce 0%
map 54% reduce 0%
map 55% reduce 0%
map 56% reduce 0%
map 57% reduce 0%
map 58% reduce 0%
map 59% reduce 0%
map 60% reduce 0%
map 64% reduce 0%
map 65% reduce 0%
map 66% reduce 0%
map 67% reduce 0%
map 68% reduce 0%
map 69% reduce 0%
map 70% reduce 0%
map 71% reduce 0%
map 75% reduce 0%
map 77% reduce 0%
map 83% reduce 0%
map 88% reduce 0%
map 90% reduce 0%
map 96% reduce 0%
map 97% reduce 0%
map 100% reduce 0%
map 100% reduce 29%
map 100% reduce 38%
map 100% reduce 45%
```

```
   map 100% reduce 52%
   map 100% reduce 58%
   map 100% reduce 64%
   map 100% reduce 67%
   map 100% reduce 68%
   map 100% reduce 69%
   map 100% reduce 70%
   map 100% reduce 71%
   map 100% reduce 72%
   map 100% reduce 73%
   map 100% reduce 74%
   map 100% reduce 75%
   map 100% reduce 76%
   map 100% reduce 77%
   map 100% reduce 78%
   map 100% reduce 79%
   map 100% reduce 80%
   map 100% reduce 81%
   map 100% reduce 82%
   map 100% reduce 83%
   map 100% reduce 84%
   map 100% reduce 85%
   map 100% reduce 86%
   map 100% reduce 87%
   map 100% reduce 88%
   map 100% reduce 89%
   map 100% reduce 90%
   map 100% reduce 91%
   map 100% reduce 92%
   map 100% reduce 93%
   map 100% reduce 94%
   map 100% reduce 95%
   map 100% reduce 96%
   map 100% reduce 97%
   map 100% reduce 98%
   map 100% reduce 99%
   map 100% reduce 100%
  Job job_1475792131280_1207 completed successfully
  Output directory: hdfs:///user/ask/tmp/mrjob/prEDA.ask.20161112.174047.514269/output
Counters: 49
        File Input Format Counters
                Bytes Read=2091377120
        File Output Format Counters
                Bytes Written=29
        File System Counters
                FILE: Number of bytes read=476928861
                FILE: Number of bytes written=754869895
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2091378488
                HDFS: Number of bytes written=29
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=27
```

```
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=8
                Launched reduce tasks=1
                Rack-local map tasks=8
                Total megabyte-milliseconds taken by all map tasks=3549700608
                Total megabyte-milliseconds taken by all reduce tasks=2703989760
                Total time spent by all map tasks (ms)=2311003
                Total time spent by all maps in occupied slots (ms)=6933009
                Total time spent by all reduce tasks (ms)=1056246
                Total time spent by all reduces in occupied slots (ms)=5281230
                Total vcore-milliseconds taken by all map tasks=2311003
                Total vcore-milliseconds taken by all reduce tasks=1056246
        Map-Reduce Framework
                CPU time spent (ms)=3746740
                Combine input records=0
                Combine output records=0
                Failed Shuffles=0
                GC time elapsed (ms)=2092
                Input split bytes=1368
                Map input records=5781290
                Map output bytes=1823173121
                Map output materialized bytes=276773298
                Map output records=147895347
                Merged Map outputs=8
                Physical memory (bytes) snapshot=8787091456
                Reduce input groups=15192277
                Reduce input records=147895347
                Reduce output records=1
                Reduce shuffle bytes=276773298
                Shuffled Maps =8
                Spilled Records=428732139
                Total committed heap usage (bytes)=14943256576
                Virtual memory (bytes) snapshot=20894380032
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
Streaming final output from hdfs:///user/ask/tmp/mrjob/prEDA.ask.20161112.174047.514269/output...
"Number of Nodes: "        15192277
Removing HDFS temp directory hdfs:///user/ask/tmp/mrjob/prEDA.ask.20161112.174047.514269...
Removing temp directory /tmp/prEDA.ask.20161112.174047.514269...
```

```
In [3]: #So obviously this failed miserably and I am sad
        !hdfs dfs -rmr wiki5Iter
        !./PageRankAlternate2.py PageRank-test_indexed.txt -r hadoop --iterations 1 --N 11 --alpha .15 -
```

```
rmr: DEPRECATED: Please use 'rm -r' instead.
16/11/14 13:52:34 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 5760 mi
Moved: 'hdfs://nn-ia.s3s.altiscale.com:8020/user/ask/wiki5Iter' to trash at: hdfs://nn-ia.s3s.altiscale
No configs found; falling back on auto-configuration
Creating temp directory /tmp/PageRankAlternate2.ask.20161114.135235.320749
```

```
Looking for hadoop binary in /opt/hadoop/bin...
Found hadoop binary: /opt/hadoop/bin/hadoop
Using Hadoop version 2.7.2
Copying local files to hdfs:///user/ask/tmp/mrjob/PageRankAlternate2.ask.20161114.135235.320749/files/.
Looking for Hadoop streaming jar in /opt/hadoop...
Found Hadoop streaming jar: /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar
Detected hadoop configuration property names that do not match hadoop version 2.7.2:
The have been translated as follows
 mapred.reduce.tasks: mapreduce.job.reduces
Running step 1 of 2...
  packageJobJar: [] [/opt/hadoop-2.7.2/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar] /tmp/streamjo
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Loaded native gpl library from the embedded binaries
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev d62701d4d05dfa6115bbaf8d9dff002d
  Total input paths to process : 1
  number of splits:2
  Submitting tokens for job: job_1475792131280_1343
  Submitted application application_1475792131280_1343
  The url to track the job: http://rm-ia.s3s.altiscale.com:8088/proxy/application_1475792131280_1343/
  Running job: job_1475792131280_1343
  Job job_1475792131280_1343 running in uber mode : false
   map 0% reduce 0%
   map 100% reduce 0%
   map 100% reduce 100%
  Job job_1475792131280_1343 completed successfully
  Output directory: hdfs:///user/ask/tmp/mrjob/PageRankAlternate2.ask.20161114.135235.320749/step-outpu
Counters: 49
        File Input Format Counters
                Bytes Read=252
        File Output Format Counters
                Bytes Written=459
        File System Counters
                FILE: Number of bytes read=165
                FILE: Number of bytes written=391675
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=622
                HDFS: Number of bytes written=459
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=9
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Rack-local map tasks=2
                Total megabyte-milliseconds taken by all map tasks=11430912
                Total megabyte-milliseconds taken by all reduce tasks=10695680
                Total time spent by all map tasks (ms)=7442
```

```
                Total time spent by all maps in occupied slots (ms)=22326
                Total time spent by all reduce tasks (ms)=4178
                Total time spent by all reduces in occupied slots (ms)=20890
                Total vcore-milliseconds taken by all map tasks=7442
                Total vcore-milliseconds taken by all reduce tasks=4178
        Map-Reduce Framework
                CPU time spent (ms)=3060
                Combine input records=27
                Combine output records=13
                Failed Shuffles=0
                GC time elapsed (ms)=83
                Input split bytes=370
                Map input records=10
                Map output bytes=307
                Map output materialized bytes=197
                Map output records=27
                Merged Map outputs=2
                Physical memory (bytes) snapshot=1881669632
                Reduce input groups=11
                Reduce input records=13
                Reduce output records=11
                Reduce shuffle bytes=197
                Shuffled Maps =2
                Spilled Records=26
                Total committed heap usage (bytes)=5218762752
                Virtual memory (bytes) snapshot=7760326656
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
Detected hadoop configuration property names that do not match hadoop version 2.7.2:
The have been translated as follows
 mapred.reduce.tasks: mapreduce.job.reduces
Running step 2 of 2...
  packageJobJar: [] [/opt/hadoop-2.7.2/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar] /tmp/streamjo
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Loaded native gpl library from the embedded binaries
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev d62701d4d05dfa6115bbaf8d9dff002d
  Total input paths to process : 1
  number of splits:2
  Submitting tokens for job: job_1475792131280_1344
  Submitted application application_1475792131280_1344
  The url to track the job: http://rm-ia.s3s.altiscale.com:8088/proxy/application_1475792131280_1344/
  Running job: job_1475792131280_1344
  Job job_1475792131280_1344 running in uber mode : false
   map 0% reduce 0%
```

```
   map 100% reduce 0%
   map 100% reduce 100%
  Job job_1475792131280_1344 completed successfully
  Output directory: hdfs:///user/ask/wiki5Iter
Counters: 49
        File Input Format Counters
                Bytes Read=689
        File Output Format Counters
                Bytes Written=456
        File System Counters
                FILE: Number of bytes read=267
                FILE: Number of bytes written=390359
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1051
                HDFS: Number of bytes written=456
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=9
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Rack-local map tasks=2
                Total megabyte-milliseconds taken by all map tasks=20398080
                Total megabyte-milliseconds taken by all reduce tasks=17740800
                Total time spent by all map tasks (ms)=13280
                Total time spent by all maps in occupied slots (ms)=39840
                Total time spent by all reduce tasks (ms)=6930
                Total time spent by all reduces in occupied slots (ms)=34650
                Total vcore-milliseconds taken by all map tasks=13280
                Total vcore-milliseconds taken by all reduce tasks=6930
        Map-Reduce Framework
                CPU time spent (ms)=3860
                Combine input records=30
                Combine output records=16
                Failed Shuffles=0
                GC time elapsed (ms)=82
                Input split bytes=362
                Map input records=11
                Map output bytes=806
                Map output materialized bytes=294
                Map output records=30
                Merged Map outputs=2
                Physical memory (bytes) snapshot=1724055552
                Reduce input groups=12
                Reduce input records=16
                Reduce output records=11
                Reduce shuffle bytes=294
                Shuffled Maps =2
                Spilled Records=32
                Total committed heap usage (bytes)=2481455104
                Virtual memory (bytes) snapshot=7746486272
        Shuffle Errors
```

```
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
Streaming final output from hdfs:///user/ask/wiki5Iter...
"1"        [{}, 0.059297520661157024]
"10"        [{"5": 1}, 0.020661157024793389]
"11"        [{"5": 1}, 0.020661157024793389]
"2"        [{"3": 1}, 0.31687327823691458]
"3"        [{"2": 1}, 0.09793388429752066]
"4"        [{"1": 1, "2": 1}, 0.046418732782369146]
"5"        [{"2": 1, "4": 1, "6": 1}, 0.32975206611570246]
"6"        [{"2": 1, "5": 1}, 0.046418732782369146]
"7"        [{"2": 1, "5": 1}, 0.020661157024793389]
"8"        [{"2": 1, "5": 1}, 0.020661157024793389]
"9"        [{"2": 1, "5": 1}, 0.020661157024793389]
Removing HDFS temp directory hdfs:///user/ask/tmp/mrjob/PageRankAlternate2.ask.20161114.135235.320749..
Removing temp directory /tmp/PageRankAlternate2.ask.20161114.135235.320749...
```

```
In [3]: %%writefile runIndividualPR.py
        #!/usr/bin/env python

        from numpy import random,array
        from PageRankScale import PageRankScale
        from PageRankPreprocess import PageRankPreprocess
        import sys

        #Plan: Preprocess -> run -> get counter ->

        #Don't forget to add -r hadoop
        loss = sys.argv[3]
        mr_job = PageRankScale(args=['hdfs:/user/ask/preprocessed'+sys.argv[1]+'/', "-r", "hadoop", "--
        print loss
        print sys.argv[1]
        print sys.argv[2]

        with mr_job.make_runner() as runner:
            runner.run()
            lossMap = runner.counters()
            loss = lossMap[0]["loss"]["lossCounter"]
            print loss


Overwriting runIndividualPR.py
```

3.3 I had numerous issues running this job from within the notebook because my connection would always break out before the job could complete (~50 minutes/iteration, 11 iterations, you see where this is going). I instead ran individual iterations in the background throughout the day so i could check status after each. I ran the below command for iters 1-10, passing the loss from the previous run's counter multiplied by some large scalar.

In [ ]:
```
./runIndividualPR.py 1 2 485328587534
```

In [8]:
```
%%writefile Sort.py
#!/usr/bin/env python

from mrjob.job import MRJob
from mrjob.step import MRStep
import ast
import bisect
import itertools
import re

class Sort(MRJob):

    def configure_options(self):
        super(Sort, self).configure_options()
        self.add_passthrough_option('--N', type = "int", default = 100)

    def __init__(self, *args, **kwargs):
        super(Sort, self).__init__(*args, **kwargs)
        self.options.jobconf = {"mapred.reduce.tasks":1 ,
                                'mapred.output.key.comparator.class':'org.apache.hadoop.mapred.
                                'mapred.text.key.comparator.options': '-k1r'}
        self.N = self.options.N
        self.nodes = ["nada" for i in range(self.N)]
        self.topN = [0 for i in range(self.N)]

    def mapper(self, node, line):
        key, value = line.strip().split("\t")
        key = key.replace('"','')
        links, prob = ast.literal_eval(value)

        yield prob, key

    def reducer(self, node, lines):
        if node > self.topN[0]:
            insertionIndex = bisect.bisect(self.topN,node)
            self.nodes.insert(insertionIndex, next(lines))
            self.topN.insert(insertionIndex, node)
            del self.nodes[0]
            del self.topN[0]

    def reducer_final(self):
        self.topN.reverse()
        self.nodes.reverse()
        for i in range(self.N):
            yield self.nodes[i] , self.topN[i]
```

```
            def steps(self):
                return (
                        [MRStep(mapper = self.mapper, reducer = self.reducer, reducer_final = self.redu
                )

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

Overwriting Sort.py

In [9]: #Testing this
        !./Sort.py "testOut.txt" --N 20

No configs found; falling back on auto-configuration
Creating temp directory /tmp/Sort.ask.20161115.232514.971957
Running step 1 of 1...
Streaming final output from /tmp/Sort.ask.20161115.232514.971957/output...
"5"        0.3227272727272727
"2"        0.3098484848484848
"3"        0.09090909090909091
"1"        0.05227272727272727
"4"        0.03939393939393939
"10"        0.013636363636363636
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
"nada"        0
Removing temp directory /tmp/Sort.ask.20161115.232514.971957...

In [10]: !./Sort.py "hdfs:/user/ask/preprocessed5/" -r hadoop --N 200 --output-dir "5IterOutput"

No configs found; falling back on auto-configuration
Looking for hadoop binary in /opt/hadoop/bin...
Found hadoop binary: /opt/hadoop/bin/hadoop
Creating temp directory /tmp/Sort.ask.20161115.232543.418190
Using Hadoop version 2.7.2
Copying local files to hdfs:///user/ask/tmp/mrjob/Sort.ask.20161115.232543.418190/files/...
Looking for Hadoop streaming jar in /opt/hadoop...
Found Hadoop streaming jar: /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar
Detected hadoop configuration property names that do not match hadoop version 2.7.2:
The have been translated as follows
 mapred.output.key.comparator.class: mapreduce.job.output.key.comparator.class
mapred.reduce.tasks: mapreduce.job.reduces
mapred.text.key.comparator.options: mapreduce.partition.keycomparator.options
```

```
Running step 1 of 1...
  packageJobJar: [] [/opt/hadoop-2.7.2/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar] /tmp/streamjo
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Loaded native gpl library from the embedded binaries
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev d62701d4d05dfa6115bbaf8d9dff002d
  Total input paths to process : 4
  number of splits:12
  Submitting tokens for job: job_1475792131280_1527
  Submitted application application_1475792131280_1527
  The url to track the job: http://rm-ia.s3s.altiscale.com:8088/proxy/application_1475792131280_1527/
  Running job: job_1475792131280_1527
  Job job_1475792131280_1527 running in uber mode : false
   map 0% reduce 0%
   map 4% reduce 0%
   map 5% reduce 0%
   map 6% reduce 0%
   map 8% reduce 0%
   map 10% reduce 0%
   map 12% reduce 0%
   map 14% reduce 0%
   map 16% reduce 0%
   map 18% reduce 0%
   map 20% reduce 0%
   map 22% reduce 0%
   map 24% reduce 0%
   map 26% reduce 0%
   map 28% reduce 0%
   map 30% reduce 0%
   map 32% reduce 0%
   map 33% reduce 0%
   map 35% reduce 0%
   map 37% reduce 0%
   map 39% reduce 0%
   map 41% reduce 0%
   map 42% reduce 0%
   map 50% reduce 0%
   map 53% reduce 0%
   map 54% reduce 0%
   map 55% reduce 0%
   map 56% reduce 0%
   map 57% reduce 0%
   map 58% reduce 0%
   map 59% reduce 0%
   map 60% reduce 0%
   map 61% reduce 0%
   map 62% reduce 0%
   map 63% reduce 0%
   map 64% reduce 0%
   map 65% reduce 0%
```

```
map 66% reduce 0%
map 67% reduce 0%
map 68% reduce 0%
map 69% reduce 0%
map 70% reduce 0%
map 71% reduce 0%
map 72% reduce 0%
map 73% reduce 0%
map 74% reduce 0%
map 75% reduce 0%
map 76% reduce 0%
map 77% reduce 0%
map 80% reduce 0%
map 83% reduce 0%
map 89% reduce 0%
map 92% reduce 0%
map 97% reduce 0%
map 100% reduce 0%
map 100% reduce 39%
map 100% reduce 43%
map 100% reduce 48%
map 100% reduce 53%
map 100% reduce 58%
map 100% reduce 65%
map 100% reduce 67%
map 100% reduce 68%
map 100% reduce 69%
map 100% reduce 70%
map 100% reduce 71%
map 100% reduce 72%
map 100% reduce 73%
map 100% reduce 74%
map 100% reduce 75%
map 100% reduce 76%
map 100% reduce 77%
map 100% reduce 78%
map 100% reduce 79%
map 100% reduce 80%
map 100% reduce 81%
map 100% reduce 82%
map 100% reduce 83%
map 100% reduce 84%
map 100% reduce 85%
map 100% reduce 86%
map 100% reduce 87%
map 100% reduce 88%
map 100% reduce 89%
map 100% reduce 90%
map 100% reduce 91%
map 100% reduce 92%
map 100% reduce 93%
map 100% reduce 94%
map 100% reduce 95%
map 100% reduce 96%
```

```
   map 100% reduce 98%
   map 100% reduce 99%
   map 100% reduce 100%
  Job job_1475792131280_1527 completed successfully
  Output directory: hdfs:///user/ask/5IterOutput
Counters: 51
        File Input Format Counters
                Bytes Read=2621593029
        File Output Format Counters
                Bytes Written=6664
        File System Counters
                FILE: Number of bytes read=244889754
                FILE: Number of bytes written=516079040
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2621594481
                HDFS: Number of bytes written=6664
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=39
                HDFS: Number of write operations=2
        Job Counters
                Data-local map tasks=2
                Killed map tasks=6
                Launched map tasks=18
                Launched reduce tasks=1
                Rack-local map tasks=16
                Total megabyte-milliseconds taken by all map tasks=2991341568
                Total megabyte-milliseconds taken by all reduce tasks=996060160
                Total time spent by all map tasks (ms)=1947488
                Total time spent by all maps in occupied slots (ms)=5842464
                Total time spent by all reduce tasks (ms)=389086
                Total time spent by all reduces in occupied slots (ms)=1945430
                Total vcore-milliseconds taken by all map tasks=1947488
                Total vcore-milliseconds taken by all reduce tasks=389086
        Map-Reduce Framework
                CPU time spent (ms)=2030770
                Combine input records=0
                Combine output records=0
                Failed Shuffles=0
                GC time elapsed (ms)=8264
                Input split bytes=1452
                Map input records=15192277
                Map output bytes=503879131
                Map output materialized bytes=269504116
                Map output records=15192277
                Merged Map outputs=12
                Physical memory (bytes) snapshot=11440603136
                Reduce input groups=7963043
                Reduce input records=15192277
                Reduce output records=200
                Reduce shuffle bytes=269504116
                Shuffled Maps =12
                Spilled Records=30384554
```

```
                Total committed heap usage (bytes)=12607553536
                Virtual memory (bytes) snapshot=29664174080
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
Streaming final output from hdfs:///user/ask/5IterOutput...
"13455888"        0.0014374537931089069
"1184351"        0.0006730283032776684 9
"4695850"        0.0006329585162204179 5
"5051368"        0.0005674249198901090 4
"1384888"        0.0004537140801411258 6
"7902219"        0.0004501627589584407 9
"6113490"        0.0004484363327375931 5
"2437837"        0.0004385887534989427 2
"6076759"        0.0004238728388350332 2
"13425865"        0.0004215314999479850 8
"4196067"        0.0004155049060811911 1
"6172466"        0.0004020236217288207 7
"14112583"        0.0003757367094482611 5
"10390714"        0.0003625257975362180 5
"15164193"        0.0003450351869540523 8
"3191491"        0.000339795834014251 7
"7835160"        0.000327312004665339 53
"6416278"        0.000325904675718274 81
"6237129"        0.000322840515874924 61
"1516699"        0.000319727986180314 67
"13725487"        0.000316073396746010 44
"7576704"        0.000310146930970070 59
"9276255"        0.000308286461217165 54
"10469541"        0.000306215803220078 38
"5154210"        0.000300564752591696 72
"7990491"        0.000275280315757411 03
"12836211"        0.000272412058733875 44
"4198751"        0.000263271185604942 69
"2797855"        0.000258680408898754 93
"11253108"        0.000257145592159747 36
"3603527"        0.000256097123327491 01
"3069099"        0.000249717899923783 71
"9386580"        0.000249130160990491 63
"12074312"        0.000247116746066039
"14881689"        0.000247033379295109 39
"2155467"        0.000243820615844517 64
"1441065"        0.000236930656881054 7
"14503460"        0.000231303358576077 28
"3191268"        0.000217654285117209 72
"10566120"        0.000217244736271867 43
"2396749"        0.000216656084326627 06
"11147327"        0.000211543432670531 61
"2614581"        0.000211464141453925 77
"1637982"        0.000209388062661247 12
```

```
"11245362"        0.00020410429537213727
"12430985"        0.00020111839727265661
"9355455"        0.00019319868743863708
"10527224"        0.00019087708849827142
"6172167"        0.00018880176596116968
"2614578"        0.00018840730315357299
"981395"        0.00018648942981411198
"8697871"        0.00018554170635723508
"14112408"        0.00018547977585372667
"9391762"        0.00018303283182244387
"6171937"        0.00018032024296396075
"5490435"        0.00017782294119787225
"14725161"        0.00017089476169998048
"11582765"        0.00016918780095277782
"9562547"        0.00016613413869562781
"994890"        0.00016522335468873749
"12067030"        0.00016276457732103958
"10345830"        0.00016067417852533492
"9394907"        0.00015817468816575427
"13280859"        0.00015706650448403918
"9997298"        0.00015639446882820818
"4978429"        0.00015501489680373192
"12447593"        0.00015209423533826867
"11148415"        0.00014926198003649324
"8019937"        0.00014901868427607336
"13432150"        0.00014707585848422334
"4344962"        0.00014439440720674557
"1175360"        0.00014301956923402928
"4624519"        0.00014141180454607498
"12038331"        0.00013953140667357267
"14565507"        0.00013833225463663601
"14981725"        0.00013605059425113181
"1523975"        0.00013387143651993032
"13328060"        0.00013321236652489766
"2826544"        0.00013073175557463447
"10399499"        0.00013011050869765655
"1332806"        0.00012952533061054715
"14963657"        0.00012883673685809162
"1813634"        0.00012418528685627942
"2578813"        0.00012363182145973961
"1575979"        0.00012252901906733614
"2778099"        0.00012042613255552958
"9924814"        0.00011892169798904968
"13853369"        0.00011540883369278532
"4568647"        0.00011452857328024157
"14727077"        0.00011420870332948459
"3328327"        0.00011285834942255314
"15070394"        0.00011283345702763463
"3973000"        0.00011259223124723228
"9742161"        0.00011198081711763843
"14709489"        0.00011191417417453082
"4320007"        0.00011152294646317119
"10246542"        0.00011150356768644128
"3591832"        0.00011147497010750559
```

```
"5908108"        0.00011106835052193648
"5274313"        0.0001110489626352598 3
"13433669"        0.00011087635811202597
"12785678"        0.00010992900472579289
"7467127"        0.0001098824111207262 5
"3972999"        0.0001082173065995878 7
"1561395"        0.0001074622874847979 8
"10447338"        0.00010699499959392211
"12685893"        0.000106533163167185
"3577363"        0.0001059203139302076 1
"12048800"        0.00010509773625626295
"9390959"        0.0001045788138638951 4
"10917716"        0.00010389279919507965
"4783981"        0.0001029706530931290 8
"8641167"        0.0001023590776115362 4
"1062600"        0.0001013225652268733 9
"1797655"        0.0001011211672487849 8
"4502743"        9.9620455785324045e-05
"10345922"        9.7722654162465215e-05
"12077147"        9.657433957845234e-05
"13383446"        9.6213861917041668e-05
"10131397"        9.5937898317655004e-05
"14281833"        9.4676263547014669e-05
"13348221"        9.4425293608012121e-05
"6089405"        9.4381204777831482e-05
"6176135"        9.4337502133794263e-05
"5045896"        9.3581383368030935e-05
"1184441"        9.3477814034126564e-05
"1053553"        9.253659190960176e-05
"14774430"        9.2182273489366332e-05
"14523062"        9.2136227704606134e-05
"6222591"        9.1874118585462191e-05
"1947095"        9.1836675889382812e-05
"14941437"        9.1080656525794601e-05
"4622124"        9.1012564020100835e-05
"14691339"        9.0603332208549691e-05
"2826542"        9.0497831005983967e-05
"14614474"        9.042857807527139e-05
"14967969"        8.8848202932290249e-05
"15140258"        8.7972084526345142e-05
"10232589"        8.7760421094148889e-05
"11673081"        8.7737860890223831e-05
"8446758"        8.7018544472374304e-05
"12530354"        8.6724145676257412e-05
"1864361"        8.6145222962886106e-05
"13802598"        8.6141171683233196e-05
"11142409"        8.6097809528172929e-05
"4502744"        8.5941964506516222e-05
"4350909"        8.5393116904262435e-05
"11150209"        8.5255874886006004e-05
"1522801"        8.5080110178907818e-05
"9534683"        8.4192895168334935e-05
"3974919"        8.3936058229933317e-05
"8765370"        8.3702279080239976e-05
```

```
"14886640"        8.2921188940253933e-05
"10169773"        8.266461446973671e-05
"14517385"        8.1498329026967196e-05
"12131440"        8.1442324635367603e-05
"10665856"        8.095069086647863e-05
"6041511"        8.0943369340550725e-05
"13135057"        8.0753516222212469e-05
"4302432"        8.0634386035618457e-05
"11449900"        8.0586375521683807e-05
"11943115"        8.0233618702642385e-05
"3036297"        8.0029980464963208e-05
"13429704"        7.876698628657928e-05
"9262529"        7.8203682960598512e-05
"13691678"        7.7777571640772191e-05
"2198277"        7.767819071320613e-05
"12453082"        7.7326575172508252e-05
"13846451"        7.7152964340563725e-05
"2268616"        7.6861309716764383e-05
"13944830"        7.6105921137993685e-05
"10916601"        7.578108166082121e-05
"9382973"        7.5641330931108989e-05
"343926"        7.5554708344016447e-05
"9010668"        7.5166917750785109e-05
"8273081"        7.5018929353887445e-05
"3245866"        7.4752326143053975e-05
"13443575"        7.4721984253682116e-05
"1518823"        7.3771459209320684e-05
"8502227"        7.3567772952843667e-05
"8019717"        7.2781736499124051e-05
"10474525"        7.2761254076133739e-05
"10728264"        7.2739149942997738e-05
"884277"        7.2704964106075356e-05
"11942606"        7.2703898054118468e-05
"12222311"        7.26879827099608e-05
"3554091"        7.2524044919330515e-05
"4056814"        7.2454185717645047e-05
"2790203"        7.2314200174723964e-05
"8258220"        7.2195601369496473e-05
"9376362"        7.094523486817472e-05
"10425498"        7.0935093638451062e-05
"8630387"        7.0727177233977158e-05
"6206562"        7.0564654490600575e-05
"11102316"        7.0468460620774532e-05
"6080621"        7.0269123847338202e-05
"11623371"        6.9714841941201331e-05
"9861483"        6.9639783291128602e-05
"5756977"        6.9514241909192207e-05
"10140404"        6.9280847515598595e-05
Removing HDFS temp directory hdfs:///user/ask/tmp/mrjob/Sort.ask.20161115.232543.418190...
Removing temp directory /tmp/Sort.ask.20161115.232543.418190...
```

```
In [1]: !./Sort.py "hdfs:/user/ask/preprocessed10/" -r hadoop --N 200 --output-dir "10IterOutput"
```

```
No configs found; falling back on auto-configuration
Looking for hadoop binary in /opt/hadoop/bin...
```

```
Found hadoop binary: /opt/hadoop/bin/hadoop
Creating temp directory /tmp/Sort.ask.20161116.033358.081150
Using Hadoop version 2.7.2
Copying local files to hdfs:///user/ask/tmp/mrjob/Sort.ask.20161116.033358.081150/files/...
Looking for Hadoop streaming jar in /opt/hadoop...
Found Hadoop streaming jar: /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar
Detected hadoop configuration property names that do not match hadoop version 2.7.2:
The have been translated as follows
 mapred.output.key.comparator.class: mapreduce.job.output.key.comparator.class
mapred.reduce.tasks: mapreduce.job.reduces
mapred.text.key.comparator.options: mapreduce.partition.keycomparator.options
Running step 1 of 1...
  packageJobJar: [] [/opt/hadoop-2.7.2/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar] /tmp/streamjo
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Timeline service address: http://rm-ia.s3s.altiscale.com:8188/ws/v1/timeline/
  Connecting to ResourceManager at rm-ia.s3s.altiscale.com/10.251.255.108:8032
  Connecting to Application History server at rm-ia.s3s.altiscale.com/10.251.255.108:10200
  Loaded native gpl library from the embedded binaries
  Successfully loaded & initialized native-lzo library [hadoop-lzo rev d62701d4d05dfa6115bbaf8d9dff002d
  Total input paths to process : 4
  number of splits:12
  Submitting tokens for job: job_1475792131280_1536
  Submitted application application_1475792131280_1536
  The url to track the job: http://rm-ia.s3s.altiscale.com:8088/proxy/application_1475792131280_1536/
  Running job: job_1475792131280_1536
  Job job_1475792131280_1536 running in uber mode : false
   map 0% reduce 0%
   map 2% reduce 0%
   map 4% reduce 0%
   map 5% reduce 0%
   map 6% reduce 0%
   map 7% reduce 0%
   map 8% reduce 0%
   map 9% reduce 0%
   map 10% reduce 0%
   map 11% reduce 0%
   map 12% reduce 0%
   map 14% reduce 0%
   map 16% reduce 0%
   map 18% reduce 0%
   map 20% reduce 0%
   map 22% reduce 0%
   map 23% reduce 0%
   map 25% reduce 0%
   map 27% reduce 0%
   map 29% reduce 0%
   map 31% reduce 0%
   map 33% reduce 0%
   map 35% reduce 0%
   map 37% reduce 0%
   map 38% reduce 0%
   map 40% reduce 0%
```

```
map 42% reduce 0%
map 47% reduce 0%
map 51% reduce 0%
map 54% reduce 0%
map 55% reduce 0%
map 56% reduce 0%
map 57% reduce 0%
map 58% reduce 0%
map 59% reduce 0%
map 60% reduce 0%
map 61% reduce 0%
map 62% reduce 0%
map 63% reduce 0%
map 64% reduce 0%
map 65% reduce 0%
map 66% reduce 0%
map 67% reduce 0%
map 68% reduce 0%
map 69% reduce 0%
map 70% reduce 0%
map 71% reduce 0%
map 72% reduce 0%
map 73% reduce 0%
map 74% reduce 0%
map 75% reduce 0%
map 76% reduce 0%
map 77% reduce 0%
map 78% reduce 0%
map 83% reduce 0%
map 86% reduce 0%
map 89% reduce 0%
map 92% reduce 0%
map 94% reduce 0%
map 100% reduce 0%
map 100% reduce 39%
map 100% reduce 43%
map 100% reduce 49%
map 100% reduce 54%
map 100% reduce 60%
map 100% reduce 67%
map 100% reduce 68%
map 100% reduce 69%
map 100% reduce 70%
map 100% reduce 71%
map 100% reduce 72%
map 100% reduce 73%
map 100% reduce 74%
map 100% reduce 75%
map 100% reduce 76%
map 100% reduce 77%
map 100% reduce 78%
map 100% reduce 79%
map 100% reduce 80%
map 100% reduce 81%
```

```
   map 100% reduce 82%
   map 100% reduce 83%
   map 100% reduce 84%
   map 100% reduce 85%
   map 100% reduce 86%
   map 100% reduce 87%
   map 100% reduce 88%
   map 100% reduce 89%
   map 100% reduce 90%
   map 100% reduce 91%
   map 100% reduce 92%
   map 100% reduce 93%
   map 100% reduce 94%
   map 100% reduce 95%
   map 100% reduce 96%
   map 100% reduce 98%
   map 100% reduce 99%
   map 100% reduce 100%
  Job job_1475792131280_1536 completed successfully
  Output directory: hdfs:///user/ask/10IterOutput
Counters: 51
        File Input Format Counters
                Bytes Read=2621543120
        File Output Format Counters
                Bytes Written=6660
        File System Counters
                FILE: Number of bytes read=244932669
                FILE: Number of bytes written=516168635
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2621544584
                HDFS: Number of bytes written=6660
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=39
                HDFS: Number of write operations=2
        Job Counters
                Data-local map tasks=1
                Killed map tasks=6
                Launched map tasks=18
                Launched reduce tasks=1
                Rack-local map tasks=17
                Total megabyte-milliseconds taken by all map tasks=3020132352
                Total megabyte-milliseconds taken by all reduce tasks=959096320
                Total time spent by all map tasks (ms)=1966232
                Total time spent by all maps in occupied slots (ms)=5898696
                Total time spent by all reduce tasks (ms)=374647
                Total time spent by all reduces in occupied slots (ms)=1873235
                Total vcore-milliseconds taken by all map tasks=1966232
                Total vcore-milliseconds taken by all reduce tasks=374647
        Map-Reduce Framework
                CPU time spent (ms)=2026310
                Combine input records=0
                Combine output records=0
```

```
                Failed Shuffles=0
                GC time elapsed (ms)=6225
                Input split bytes=1464
                Map input records=15192277
                Map output bytes=503829222
                Map output materialized bytes=269550770
                Map output records=15192277
                Merged Map outputs=12
                Physical memory (bytes) snapshot=11271462912
                Reduce input groups=7963202
                Reduce input records=15192277
                Reduce output records=200
                Reduce shuffle bytes=269550770
                Shuffled Maps =12
                Spilled Records=30384554
                Total committed heap usage (bytes)=12611223552
                Virtual memory (bytes) snapshot=29672218624
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
Streaming final output from hdfs:///user/ask/10IterOutput...
"13455888"        0.0014510738333933925
"1184351"        0.00066250705355724979
"4695850"        0.00063541560108851724
"5051368"        0.00057088466835824919
"1384888"        0.00044780894896005692
"2437837"        0.00044343774015461666
"6113490"        0.00044234120040856015
"7902219"        0.00044111051634301027
"13425865"        0.00042969167872750004
"6076759"        0.00042504919419679752
"4196067"        0.00042034752671656853
"6172466"        0.00039557499842266306
"14112583"        0.00038250700119111711
"10390714"        0.00036060827587769406
"15164193"        0.00034175902190754996
"3191491"        0.00033625239111094777
"6416278"        0.00032708864416366074
"6237129"        0.00032663774495658032
"7835160"        0.00032438744415768528
"1516699"        0.0003227844748258162
"13725487"        0.00031111211997265675
"9276255"        0.00030776498985136632
"7576704"        0.00030639503573317862
"10469541"        0.00030159412302065356
"5154210"        0.00029605054425472843
"12836211"        0.00028337270283213889
"7990491"        0.0002813487278178894
"4198751"        0.00026705099912914602
"2797855"        0.00026205148020484032
```

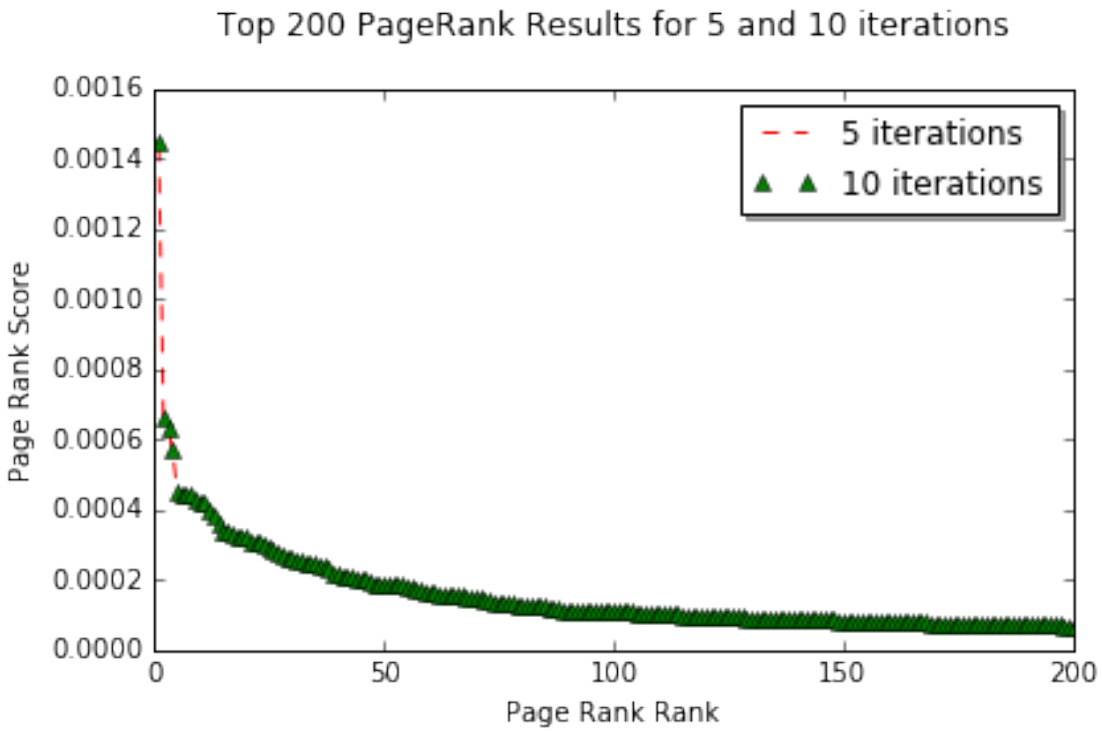
```
"11253108"        0.00025922684130908586
"9386580"        0.000255553217205677
"3603527"        0.00025348154000663029
"12074312"        0.00024928292997364792
"3069099"        0.00024723191646998426
"14881689"        0.0002441051395517641
"2155467"        0.00024321533089901982
"1441065"        0.00023712289465976067
"14503460"        0.00023179790049105903
"2396749"        0.00021904777362659168
"3191268"        0.00021384674985273355
"10566120"        0.00021343816072140449
"2614581"        0.00020995033029002453
"11147327"        0.00020993382127972008
"1637982"        0.00020596550831841252
"12430985"        0.00020195864358208459
"11245362"        0.00020148149131769464
"9355455"        0.00019558285906559841
"10527224"        0.00019042841559218417
"14112408"        0.00018927480246596994
"2614578"        0.00018691264461193568
"9391762"        0.00018668730217071014
"8697871"        0.00018581339786829783
"6172167"        0.00018577824029072465
"981395"        0.00018402610767439084
"6171937"        0.00017783754039340014
"5490435"        0.00017739935275306362
"11582765"        0.00017200208372391128
"14725161"        0.00016856511019986358
"12067030"        0.00016635245622160583
"9562547"        0.00016615384417881139
"994890"        0.00016443640913106134
"9997298"        0.00015946074059195904
"9394907"        0.00015940908868275879
"13280859"        0.00015792870656646 15
"10345830"        0.00015673116897040593
"4978429"        0.00015425816958697874
"12447593"        0.00015381987975796027
"8019937"        0.00015203260576413574
"11148415"        0.00014807815506344912
"13432150"        0.00014677336700105324
"4344962"        0.00014599341411085831
"1175360"        0.00014410832333352817
"12038331"        0.00014034463518692637
"14565507"        0.00013818301167600881
"4624519"        0.00013684824920803831
"1523975"        0.00013529251755389 54
"14981725"        0.00013416229108170933
"13328060"        0.00013376347749026991
"1332806"        0.00012983858619124245
"10399499"        0.00012939718214353326
"14963657"        0.00012911549477599588
"2826544"        0.00012745851846773 47
"2578813"        0.00012726119567882006
```

```
"1575979"        0.0001262155039358575
"1813634"        0.0001261915389467921
"2778099"        0.0001230765177400964 1
"13853369"        0.0001198375902356396 7
"9924814"        0.0001194648197042525 5
"4568647"        0.0001150113484986659 5
"12785678"        0.0001135024823139351 5
"7467127"        0.0001134623393052638 4
"9742161"        0.0001134392363718894 9
"3328327"        0.0001128652229130083 9
"10246542"        0.0001124790737690656 2
"3591832"        0.0001124439403268389 6
"5274313"        0.000112372891315504
"14727077"        0.0001123103917154245 3
"14709489"        0.0001117154864771746 5
"3973000"        0.0001114941149306094 6
"5908108"        0.0001114439022934995 7
"15070394"        0.0001112907591199491 1
"13433669"        0.0001104359547493262 9
"4320007"        0.0001099018999802949 6
"10447338"        0.0001083005122854532 6
"3577363"        0.0001079508502209357 4
"9390959"        0.0001076461183612587 9
"12685893"        0.0001075293696532473 8
"3972999"        0.0001071448770104249 2
"1561395"        0.0001060199080965739 2
"10917716"        0.0001056689814280394 1
"12048800"        0.0001053485072229400 2
"4783981"        0.0001051728624454665 9
"8641167"        0.0001040516770483992 9
"1062600"        0.0001023324813254496 5
"4502743"        9.945663466623303 8e-05
"1797655"        9.931996226437174 3e-05
"12077147"        9.787176211280885 4e-05
"10131397"        9.734959977305045 8e-05
"13383446"        9.702803904340965 3e-05
"10345922"        9.637610822984087 5e-05
"6176135"        9.581866189051575 4e-05
"6089405"        9.524780255370345 7e-05
"13348221"        9.508431662899419 6e-05
"5045896"        9.499206154704778 7e-05
"14281833"        9.463623444408073 3e-05
"1053553"        9.446674126968757 2e-05
"6222591"        9.366746050197506 3e-05
"14774430"        9.328826348697738 6e-05
"1184441"        9.240174358322136 2e-05
"4622124"        9.178308638787696 7e-05
"1947095"        9.168029629594759 2e-05
"14941437"        9.093558449226014 3e-05
"14523062"        9.06574594704229e-05
"14614474"        9.061699058422524 4e-05
"10232589"        8.993907415810513 3e-05
"2826542"        8.947771295775515 4e-05
"14691339"        8.903870680997823 2e-05
```

```
"14967969"        8.8681183731248816e-05
"12530354"        8.8362621347575515e-05
"11673081"        8.7996667582468475e-05
"1864361"        8.7836895699612249e-05
"11142409"        8.7581220934325739e-05
"13802598"        8.7545936128074746e-05
"11150209"        8.7409582423057038e-05
"4350909"        8.7296614821952517e-05
"15140258"        8.6989630541757537e-05
"8446758"        8.6820225978980996e-05
"1522801"        8.5565442450449622e-05
"4502744"        8.5155606521367981e-05
"9534683"        8.4953988002052876e-05
"3974919"        8.3883197075588443e-05
"8765370"        8.346589473211365e-05
"10169773"        8.2696388126856632e-05
"14886640"        8.2616733667363742e-05
"14517385"        8.2427232096423869e-05
"11449900"        8.2410247416722369e-05
"13135057"        8.2363738096318017e-05
"10665856"        8.2187341543220468e-05
"6041511"        8.203345886626533e-05
"12131440"        8.1268453250641468e-05
"13429704"        8.0607072738293534e-05
"4302432"        8.0515174463748838e-05
"3036297"        8.0457928992558491e-05
"9262529"        7.9916282925652753e-05
"11943115"        7.9773399058000996e-05
"13691678"        7.8902293252485146e-05
"12453082"        7.8419814703189998e-05
"2198277"        7.8305631216723899e-05
"10916601"        7.7493295706286455e-05
"2268616"        7.7236300029559252e-05
"13846451"        7.7134547308366919e-05
"9382973"        7.6646005194902855e-05
"9010668"        7.6294368486161328e-05
"13944830"        7.6060144936116155e-05
"343926"        7.5689691537722892e-05
"8273081"        7.5624657963622408e-05
"8019717"        7.5385994857735063e-05
"13443575"        7.537702296057245e-05
"3245866"        7.5085290661269403e-05
"8502227"        7.504342084491185e-05
"4056814"        7.4253449030287015e-05
"1518823"        7.4188810032658911e-05
"8258220"        7.3543618607570195e-05
"10474525"        7.3093354960466144e-05
"11942606"        7.2842671040985356e-05
"2790203"        7.2722825567508588e-05
"10728264"        7.2709755826890805e-05
"3554091"        7.2377909151862492e-05
"12222311"        7.2320918889095279e-05
"9376362"        7.2172712999905956e-05
"6206562"        7.2162570437128419e-05
```

```
"884277"        7.2140059029047557e-05
"8630387"        7.1784517154525152e-05
"9861483"        7.1562470842573132e-05
"11623371"        7.1149189132159606e-05
"10425498"        7.0523118659799518e-05
"10140404"        7.0366365220618759e-05
"11102316"        7.020571066611196e-05
"5820606"        6.9999640596364493e-05
"6080621"        6.9908228981105353e-05
Removing HDFS temp directory hdfs:///user/ask/tmp/mrjob/Sort.ask.20161116.033358.081150...
Removing temp directory /tmp/Sort.ask.20161116.033358.081150...
```

In [2]:
```
%matplotlib inline
import matplotlib.pyplot as plt

pr5 = []
pr10 = []
x = range(1, 201)

with open("5IterOutput/part-00000", "r") as f:
    for line in f.readlines():
        node, prob = line.strip().split('\t')
        pr5.append(float(prob))

with open("10IterOutput/part-00000", "r") as f:
    for line in f.readlines():
        node, prob = line.strip().split('\t')
        pr10.append(float(prob))

fig, ax = plt.subplots()
ax.plot(x, pr5, 'r--', label='5 iterations')
ax.plot(x, pr10, 'g^', label='10 iterations')
ax.set_title("Top 200 PageRank Results for 5 and 10 iterations\n")
ax.set_ylabel('Page Rank Score')
ax.set_xlabel('Page Rank Rank')

# Now add the legend with some customizations.
legend = ax.legend(loc='upper right', shadow=True)


plt.tight_layout()
plt.show()
```



3.4 The page rank is effectively the same when comparing 5 iterations with 10 iterations.

HW 9.3 Analysis

3.5 3. HW9.4: Topic-specific PageRank implementation using MRJob

Back to Table of Contents

Modify your PageRank implementation to produce a topic specific PageRank implementation, as described in:

http://www-cs-students.stanford.edu/~taherh/papers/topic-sensitive-pagerank.pdf

Note in this article that there is a special caveat to ensure that the transition matrix is irreducible. This caveat lies in footnote 3 on page 3:

```
A minor caveat: to ensure that M is irreducible when p
contains any 0 entries, nodes not reachable from nonzero
nodes in p should be removed. In practice this is not problematic.
```

and must be adhered to for convergence to be guaranteed.

Run topic specific PageRank on the following randomly generated network of 100 nodes:

s3://ucb-mids-mls-networks/randNet.txt (also available on Dropbox)

which are organized into ten topics, as described in the file:

s3://ucb-mids-mls-networks/randNet_topics.txt (also available on Dropbox)

Since there are 10 topics, your result should be 11 PageRank vectors (one for the vanilla PageRank implementation in 9.1, and one for each topic with the topic specific implementation). Print out the top ten ranking nodes and their topics for each of the 11 versions, and comment on your result. Assume a teleportation factor of 0.15 in all your analyses.

One final and important comment here: please consider the requirements for irreducibility with topic-specific PageRank. In particular, the literature ensures irreducibility by requiring that nodes not reachable from in-topic nodes be removed from the network.

This is not a small task, especially as it it must be performed separately for each of the (10) topics.

So, instead of using this method for irreducibility, please comment on why the literature's method is difficult to implement, and what what extra computation it will require.

Then for your code, please use the alternative, non-uniform damping vector:

```
vji = beta*(1/|Tj|); if node i lies in topic Tj

vji = (1-beta)*(1/(N - |Tj|)); if node i lies outside of topic Tj
```

for beta in (0,1) close to 1.

With this approach, you will not have to delete any nodes. If beta > 0.5, PageRank is topic-sensitive, and if beta < 0.5, the PageRank is anti-topic-sensitive. For any value of beta irreducibility should hold, so please try beta=0.99, and perhaps some other values locally, on the smaller networks.

HW 9.4 Implementation

```
In [10]: from collections import defaultdict
         import json

         topicCounts = defaultdict(int)
         f = open("randNet_topics.txt","r")
         all = f.readlines()

         for line in all:
             node, topic = line.strip().split('\t')
             topicCounts[topic]+=1

         f.close()

         with open("topicCounts.txt","w") as f:
             json.dump(topicCounts,f)


In [11]: %%writefile PageRankTopic.py
         #!/usr/bin/env python
         ##!/Users/AnthonySpalvieriKruse/anaconda/bin/python


         from mrjob.job import MRJob
         from mrjob.step import MRStep
         from collections import defaultdict
         import ast
         import json

         class PageRankTopic(MRJob):

             def configure_options(self):
                 super(PageRankTopic, self).configure_options()
                 self.add_passthrough_option('--iterations', type = "int", default=1)
```

```
        self.add_passthrough_option('--N', type = "int", default = -1)
        self.add_passthrough_option('--alpha', type = "float", default = .15)
        self.add_passthrough_option('--numTopics', type = "int", default = 0)
        self.add_passthrough_option('--beta', type = "float", default = .99)

    def __init__(self, *args, **kwargs):
        super(PageRankTopic, self).__init__(*args, **kwargs)
        self.iterations = self.options.iterations
        self.N = self.options.N
        self.alpha = self.options.alpha
        self.beta = self.options.beta
        self.numTopics = self.options.numTopics
        self.loss = [0]*(1+self.numTopics)
        self.options.jobconf = {"mapred.reduce.tasks":1 }
        self.nodeTopics = defaultdict(int)
        with open("topicCounts.txt", "r") as f:
            self.topicCounts = json.load(f)
        with open("randNet_topics.txt","r") as g:
            for i in g:
                node, topic = i.strip().split('\t')
                self.nodeTopics[node]=topic

    def initialize_graph_map(self, _, line):
        node, links = line.strip().split("\t")
        links = ast.literal_eval(links)
        for link in links:
            yield link, {}
        yield node, links

    def initialize_graph_combine(self, node, lines):
        links = {}
        for item in lines:
            if len(item)!=0:
                links = item
        yield node, links

    def initialize_graph_reduce(self, node, lines):
        links = {}
        for item in lines:
            if len(item)!=0:
                links = item
        yield node, (links, [float(1)/self.N]*(1+self.numTopics))

    def pass_mass(self, node, line):
        links, topicProbs = line
        #Gets called on the first iteration
        if isinstance(links, basestring):
            links = ast.literal_eval(links)

        numLinks = len(links)
        if numLinks == 0:
            self.loss = [x + y for x, y in zip(self.loss, topicProbs)] #prob * self.inflator
            yield node, self.loss
        else:
```

```
        for link in links:
            yield link, ({}, [float(prob)/numLinks for prob in topicProbs])
        yield node, (links, [0]*(1+self.numTopics))

    def mapper_final(self):
        yield "0", (self.loss)

    def combiner(self, node, lines):
        if node == "0":
            loss = [sum(x) for x in zip(*lines)] #sum(lines)
            yield node, loss
        else:
            nodeProb = [0]*(1+self.numTopics)
            links = {}
            for item in lines:
                link, passedMass = item
                nodeProb = [x + y for x, y in zip(nodeProb, passedMass)]
                if len(link)!=0:
                    links = link
            yield node, (links, nodeProb)

    def sum_mass(self, node, lines):
        if node == "0":
            self.loss = [sum(x) for x in zip(*lines)] #sum(lines)
            #yield "lossNow", self.loss
        else:
            total=[0]*(1+self.numTopics)
            links = {}
            newProbs = []
            G = self.N
            a = self.alpha
            for item in lines:
                link, prob = item
                total= [x + y for x, y in zip(total, prob)]
                if len(link) != 0:
                    links = link


            for index, value in enumerate(total):
                if str(1+index) == self.nodeTopics[node]:
                    newProbs.append(a * (float(self.beta)/self.topicCounts[self.nodeTopics[nod
                else:
                    if (index+1) == len(total):
                        newProbs.append(float(a)/G + (1-a) * (float(self.loss[index])/G + valu
                    else:
                        newProbs.append(a * (float(1-self.beta)/(G-self.topicCounts[self.nodeT

            yield node, (links, newProbs)

    def steps(self):
        return (
                [MRStep(mapper = self.initialize_graph_map,
                        combiner = self.initialize_graph_combine,
                        reducer = self.initialize_graph_reduce)] +
```

```
                        [MRStep(mapper = self.pass_mass, mapper_final = self.mapper_final,
                                combiner=self.combiner,
                                reducer = self.sum_mass)] * self.options.iterations
                )

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

Writing PageRankTopic.py

In [13]: !./PageRankTopic.py randNet.txt --file randNet_topics.txt --file topicCounts.txt --numTopics 1

No configs found; falling back on auto-configuration
Creating temp directory /tmp/PageRankTopic.ask.20161116.001941.909262
Running step 1 of 16...
Running step 2 of 16...
Running step 3 of 16...
Running step 4 of 16...
Running step 5 of 16...
Running step 6 of 16...
Running step 7 of 16...
Running step 8 of 16...
Running step 9 of 16...
Running step 10 of 16...
Running step 11 of 16...
Running step 12 of 16...
Running step 13 of 16...
Running step 14 of 16...
Running step 15 of 16...
Running step 16 of 16...
Streaming final output from /tmp/PageRankTopic.ask.20161116.001941.909262/output...
Removing temp directory /tmp/PageRankTopic.ask.20161116.001941.909262...
```

4 Response

One big challenge to removing nodes is actually finding which nodes aren't reachable from in-topic nodes for every given topic. Nodes can be N degrees away from a topic node and still be reachable, but to actually learn this would require a graph traversal step such as djikstra's algorithm, and this would need to be done for every single topic. After this is done, the actual page rank algorithm would have to perform a totally different matrix multiplication for each topic, because every topic would have a different set of nodes removed. To achieve this, one would either need to run the job separately for each topic, or throw in a much heavier set of data structures to track which nodes to pass mass to/from and which nodes are considered dangling for each topic.

```
In [14]: from collections import defaultdict
         import ast

         topTopics = defaultdict(list)

         with open("topTopics.txt", 'r') as f:
             for line in f:
                 topic, stuff = line.strip().split('\t')
                 linksAndTopicProbs = ast.literal_eval(stuff)
                 probs = linksAndTopicProbs[1]
```

```
                for index, value in enumerate(probs):
                    topTopics[index].append((value, topic))

        for i in topTopics:
            print "Topic: ", i+1
            tops= sorted(topTopics[i], key=lambda x: x[0], reverse=True)[:10]
            for top in tops:
                print top[1], ": ", top[0]
            print
```

```
Topic:  1
"77" :  0.0324672831293
"52" :  0.0321010490015
"32" :  0.0317767200993
"85" :  0.0316501838508
"92" :  0.0302899221786
"74" :  0.0282039922071
"9" :  0.0281214261966
"15" :  0.0280549298581
"58" :  0.0279760619306
"10" :  0.027714551672

Topic:  2
"58" :  0.0431164664247
"9" :  0.0410543347742
"71" :  0.0406183353479
"73" :  0.0378626648474
"12" :  0.0334436275773
"59" :  0.0312805408723
"75" :  0.0308579497468
"52" :  0.0277355083776
"100" :  0.027572518767
"15" :  0.0274253591299

Topic:  3
"15" :  0.0442424533482
"70" :  0.0372892062425
"86" :  0.0343695212242
"91" :  0.0324908815667
"2" :  0.0313240557468
"66" :  0.0307392792761
"31" :  0.0292756555472
"40" :  0.0279477641115
"9" :  0.0275941089556
"100" :  0.0274883646689

Topic:  4
"63" :  0.0393500366283
"15" :  0.0317966214682
"83" :  0.0298829568496
"65" :  0.0293454614375
"71" :  0.0289297852851
"41" :  0.0287521630581
"85" :  0.0287028124586
```

```
"78" :  0.0286971352838
"84" :  0.0283563418489
"88" :  0.0278936114756

Topic:  5
"90" :  0.0380538023546
"99" :  0.0377574286161
"88" :  0.0368717959215
"51" :  0.0355172854205
"45" :  0.0334794208756
"34" :  0.0305496313397
"4" :  0.0297679814117
"15" :  0.0297305826308
"80" :  0.029560446948
"100" :  0.0292547176541

Topic:  6
"13" :  0.0450320618424
"56" :  0.0414938144765
"37" :  0.0388557641886
"11" :  0.0380622672238
"69" :  0.0358621527011
"23" :  0.0325981386014
"15" :  0.0311097589529
"85" :  0.0301416668014
"52" :  0.028231945868
"74" :  0.0270579461131

Topic:  7
"85" :  0.0375952614914
"25" :  0.0359622952905
"53" :  0.033480646618
"28" :  0.0327579556024
"35" :  0.0321195348546
"97" :  0.0305278847152
"55" :  0.0303013789953
"15" :  0.0285523940666
"30" :  0.0283320176459
"47" :  0.0281169862403

Topic:  8
"100" :  0.0446616717652
"61" :  0.0388420397202
"39" :  0.0359446599326
"8" :  0.0341468503055
"62" :  0.0329988063034
"87" :  0.0314590572293
"6" :  0.0303399837968
"54" :  0.0293551867941
"9" :  0.0272435216932
"58" :  0.0259017038687

Topic:  9
"94" :  0.0381858174265
```

```
"14" :  0.0366091018772
"42" :  0.0358910396434
"21" :  0.0357607617113
"57" :  0.0340344173826
"24" :  0.0305273785845
"96" :  0.0305201671735
"63" :  0.0284032330011
"61" :  0.0282855810384
"15" :  0.027451193682

Topic:  10
"74" :  0.0411029767259
"17" :  0.0355674084901
"49" :  0.0349446036957
"15" :  0.030663280799
"95" :  0.0305286323743
"63" :  0.0303481009686
"61" :  0.0297065302845
"43" :  0.0280461966565
"100" :  0.0280397204422
"85" :  0.0274582112136

Topic:  11
"15" :  0.148693846006
"74" :  0.145174445532
"63" :  0.143372186509
"100" :  0.139786500745
"85" :  0.137986582194
"9" :  0.136659199521
"58" :  0.134801535324
"71" :  0.131735143623
"61" :  0.130973157198
"52" :  0.130100358286
```

HW 9.4 Analysis
——- END OF HWK 9 ——