МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

Лабораторна робота № 2

з курсу «Оброблення надвеликих масивів даних»

Тема: «PageRank algorithm»

Виконав:

студент I курсу

групи ДА-71мн

Колінько Анжела

Київ – 2017

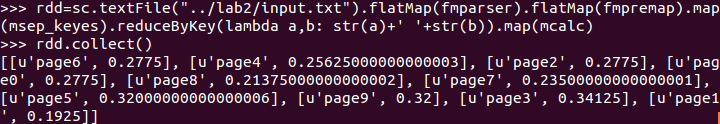
Варіант 2

1. Формула для розрахунку:

We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

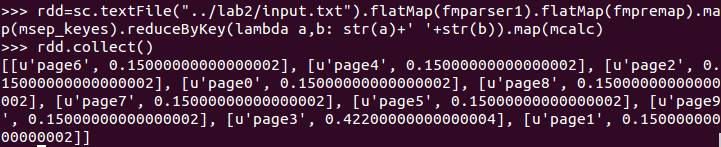
2. Для заданого файлу за першої ітерації отримаємо (спочатку всі PR =0.15):



3. Записуємо результати в файл (outp\_f.txt):

page7 0.235 page6 0.2775 page4 0.25625 page5 0.32 page2 0.2775 page9 0.32 page0 0.2775 page3 0.34125 page8 0.21375 page1 0.1925

4. За наступної ітерації:



5. Вміст файлу (outp\_f.txt):

page6 0.15 page7 0.15 page5 0.15 page4 0.15 page9 0.15 page2 0.15 page3 0.422 page0 0.15 page1 0.15 page8 0.15

Висновок: на лабораторній роботі було розглянуто PageRank algorithm

Додаток 1. lab2.py

#!/usr/bin/env python

from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("lab2.py")

sc = SparkContext(conf=conf)

d=0.85

strt\_pr=(1-d) #d=0.85 => strt\_pr=(1-d)=(1-0.85)=0.15;

bc\_strt\_pr=sc.broadcast(strt\_pr)

N=1

bc\_N=sc.broadcast(N)

def fmparser(line):

n\_links=len(line.split())-1# 1st word is a page name

line+=' '+str(bc\_strt\_pr.value)+' '+str(n\_links)

return [line]

def fmparser1(line):

n\_links=len(line.split())-1# 1st word is a page name

cr=0.; i=0;

with open("../lab2/outp\_f.txt","rb") as fin:

for fline in fin:

for fword in fline.split():

#print "fword=%s, i=%d" % (fword, i)

if(fword == (line.split())[0]): #find fword whuch name in given page name;

cr=float((fline.split())[i+1])

break;

i+=1;

fin.close()

line+=' '+str(cr)+' '+str(n\_links)

return [line]

def fmpremap(line):

arr=line.split()

ret\_line='';

for word in arr:

if(word!=arr[0] and word!=arr[len(arr)-2] and word!=arr[len(arr)-1]):#skip self links and numbers

ret\_line+=word+'\t'+arr[len(arr)-2]+' '+arr[len(arr)-1]+'\_\_\_'

#print 'len(ret\_line=%d)' % len(ret\_line)

ret\_line=ret\_line[0:len(ret\_line)-3] # deleting '\_\_\_' separator in the end of line;

return ret\_line.split('\_\_\_')

def msep\_keyes(line):

l2map=line.split('\t')

with open("../lab2/outp\_f.txt","wb") as fout:# erasing file's content to put new values in it;

pass

fout.close()

return [l2map[0], l2map[1]]

def mcalc(inp\_tuple):

page=inp\_tuple[0]

arr\_nums=inp\_tuple[1].split()

i=0; tot=0.;

for num in arr\_nums:

if not (i%2):

tot+=(float(arr\_nums[i]))/(float(arr\_nums[i+1]))

#print 'tot=%f, float(arr\_nums[%d])=%f, float(arr\_nums[%d])=%f' % (tot,i,float(arr\_nums[i]),(i+1),float(arr\_nums[i+1]))

i=i+1

tot=tot\*(1-bc\_strt\_pr.value) + bc\_strt\_pr.value

with open("../lab2/outp\_f.txt","a+b") as fout:

fout.write(str(page)+' '+str(tot)+' ')

fout.close()

return [page, tot]

rdd=sc.textFile("../lab2/input.txt").flatMap(fmparser).flatMap(fmpremap).map(msep\_keyes).reduceByKey(lambda a,b: str(a)+' '+str(b)).map(mcalc)

rdd.collect()

for itr in range(N):

rdd=sc.textFile("../lab2/input.txt").flatMap(fmparser1).flatMap(fmpremap).map(msep\_keyes).reduceByKey(lambda a,b: str(a)+' '+str(b)).map(mcalc)

rdd=sc.textFile("../lab2/input.txt").flatMap(fmparser1).flatMap(fmpremap).map(msep\_keyes).reduceByKey(lambda a,b: str(a)+' '+str(b)).map(mcalc)

rdd.collect()

Додаток 2. input.txt

page0 page1 page5 page7

page1 page0 page4

page2 page0 page3

page3 page3 page5

page4 page8 page2

page5 page3

page6 page9 page4 page7

page7 page9

page8 page6

page9 page5 page2