МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА
Факультет информатики и систем управления
Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №910
по курсу «Информационный поиск»

«Page Rank»

Выполнил:
студент группы ИУ9-21М
Беляев А. В.

Проверила:
Лукашевич Н. В.

Москва 2019

1 Цель работы

Вычислить PageRank для сетей на Рис. 1 и 2.

В ходе работы необходимо перемножать вектор исходных состояний (которые равновероятны) и матрицу переходов до схождения, либо итеративно N раз. В ходе работы был реализован алгоритм на N итераций.

Коэффициент телепортации = 0.1

2 Текст программы

```
import numpy as np

np.set_printoptions(precision=3)

ITERATIONS = 5  # it doesnt matter how many iterations
TELEPORT = 0.1


def make_transition_matrix(N, matrix):
    i = 0
    while i < len(matrix):
        row = matrix[i]

        num_of_links = row.count(1)
        if 0 != num_of_links:
            # Нормализовать в строке единицы, поделив на кол-во единиц
            row = [x / num_of_links for x in row]
            # Единицы умножить на коэффициент сглаживания (1-d)
            row = [x * (1 - TELEPORT) for x in row]
            # Ко всем элементам добавить коэффициент (d/N)
            row = [x + (TELEPORT / N) for x in row]
        else:
            # Если со страницы не было ссылок, столбцам ставим 1/N
            row = [1 / N for x in row]

        matrix[i] = row
        i += 1
    return matrix


def pagerank(N: int, matrix: list):
    vector = [1] * N
    v = np.array([x / len(vector) for x in vector])

    matrix = np.array(matrix)
    i = 0
```

```
    while i < ITERATIONS:
        v = np.matmul(v, matrix)
        i += 1
    return v


def lab9():
    N_states = 3
    init_matrix = [[0, 1, 1],
                   [0, 0, 1],
                   [0, 1, 0]]
    print('\ninit matrix:')
    print(np.array(init_matrix))

    transition_matrix = make_transition_matrix(N_states, init_matrix)
    print('\ntransition matrix:')
    print(np.array(transition_matrix))

    pr = pagerank(N_states, transition_matrix)
    print('\npagerank:')
    print(np.array(pr))


def lab10():
    N_states = 8
    init_matrix = [[0, 1, 1, 1, 0, 0, 0, 0],  # home
                   [1, 0, 0, 0, 0, 0, 0, 0],  # about
                   [1, 0, 0, 0, 0, 0, 0, 0],  # prod
                   [1, 0, 0, 0, 1, 1, 1, 1],  # links
                   [0, 0, 0, 0, 0, 0, 0, 0],  # ext A
                   [0, 0, 0, 0, 0, 0, 0, 0],  # ext B
                   [0, 0, 0, 0, 0, 0, 0, 0],  # ext C
                   [0, 0, 0, 0, 0, 0, 0, 0]]  # ext D
    print('\ninit matrix:')
    print(np.array(init_matrix))

    transition_matrix = make_transition_matrix(N_states, init_matrix)
    print('\ntransition matrix:')
    print(np.array(transition_matrix))

    pr = pagerank(N_states, transition_matrix)
    print('\npagerank:')
    print(np.array(pr))
```

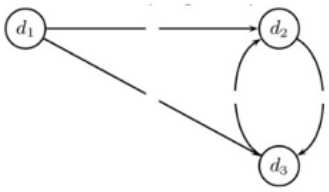

Рис. 1: Л.Р.9

```
82 if __name__ == '__main__':
83     lab9()
84     lab10()
```

3 Результаты работы

В ходе работы программы были получены следующие матрицы переходов и значения PageRank в Л.Р.9:

```
init matrix:
[[0 1 1]
 [0 0 1]
 [0 1 0]]

transition matrix:
[[0.033 0.483 0.483]
 [0.033 0.033 0.933]
 [0.033 0.933 0.033]]

pagerank:
[0.033 0.483 0.483]
```

Для Л.Р.10 результаты следующие:

```
init matrix:
[[0 1 1 1 0 0 0 0]
 [1 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0]
 [1 0 0 0 1 1 1 1]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]

transition matrix:
[[0.013 0.312 0.312 0.312 0.013 0.013 0.013 0.013]
```

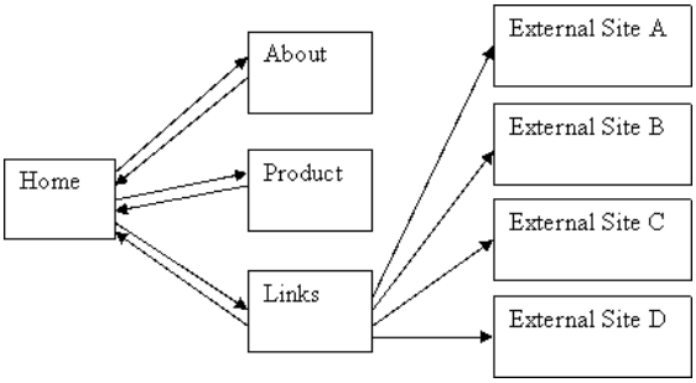

Рис. 2: Л.Р.10

```
 [0.912 0.013 0.013 0.013 0.013 0.013 0.013 0.013]
 [0.912 0.013 0.013 0.013 0.013 0.013 0.013 0.013]
 [0.193 0.013 0.013 0.013 0.193 0.193 0.193 0.193]
 [0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125]
 [0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125]
 [0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125]
 [0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125]]

pagerank:
[0.333 0.13  0.13  0.13  0.069 0.069 0.069 0.069]
```

4 Выводы

В ходе работы была изучена и реализована популярная моедль ранжирования – PageRank.