EE412 Foundation of Big Data Analytics, Fall 2019
HW3

Name: 오승준

Student ID: 20160381

Discussion Group (People with whom you discussed ideas used in your answers):
손채연 (Remove distinct input in problem 1-b, optimize problem 2-b, initialize parameter in problem 3-b)

On-line or hardcopy documents used as part of your answers:

Answer to Problem 1

(a) Solve the following problems.

- Exercise 5.1.2

  [[0.25925926]
  [0.30864198]
  [0.43209877]]

Used Python3 code for Ex-5.1.2

```
import numpy as np

M = np.array([1/3, 1/2, 0, 1/3, 0, 1/2, 1/3, 1/2, 1/2]).reshape((3,3))
M = np.identity(3) - 0.8*M
e = 0.2*np.array([1/3, 1/3, 1/3]).reshape((3,1))    # Ex 5.1.2
print(np.dot(np.linalg.inv(M), e))
```

- Exercise 5.3.1
  a. A only

     [[0.42857143]
     [0.19047619]
     [0.19047619]
     [0.19047619]]

  b. A and C

     [[0.38571429]
     [0.17142857]
     [0.27142857]
     [0.17142857]]

Used Python3 code for Ex-5.3.1

```
import numpy as np

M = np.array([0, 1/2, 1, 0, 1/3, 0, 0, 1/2, 1/3, 0, 0, 1/2, 1/3, 1/2, 0,
0]).reshape((4,4))
M = np.identity(4) - 0.8*M
e = 0.2*np.array([1, 0, 0, 0]).reshape((4,1))      # For a: A only
#e = 0.2*np.array([1/2, 0, 1/2, 0]).reshape((4,1))   # For b: A and C
print(np.dot(np.linalg.inv(M), e))
```

(b) Implement the PageRank algorithm using Spark.

Output

263	0.00216
537	0.00212
965	0.00206
243	0.00197
255	0.00194
285	0.00193
16	0.00191
126	0.00190
747	0.00190
736	0.00189

Answer to Problem 2

(a) Solve the following problems.

- Exercise 10.3.2

  a. n = 20 and d = 5

  $$20 \times \binom{5}{t} / \binom{20}{\mathrm{t}} \geq s$$

  For maximal, t = 1 and s = 5.

  b. n = 200 and d = 150

  $$200 \times \binom{150}{t} / \binom{200}{\mathrm{t}} \geq s$$

  For maximal, t = 10 and s = 10.

- Exercise 10.5.2

  a. C = {w, x}; D = {y, z}
  $P_C = 1, P_D = 1$

  b. C = {w, x, y, x}; D = {x, y, z}

  $$P_C = {}^{4}\!/\!_{\binom{4}{2}}, P_D = {}^{2}\!/\!_{\binom{3}{2}}$$

  $P_C = 2/3, P_D = 2/3$

(b) Implement the Girvan-Newman algorithm using Spark.

Output

```
764     1472    479194.22092
1020    1472    415086.77907
1020    2416    392622.59987
121     191     342187.71871
1282    1472    302374.57956
121     330     301472.86530
178     2816    301040.63715
41      764     300260.08217
330     1285    273018.58705
330     1020    271259.87615
```

Answer to Problem 3

(a) Exercise 12.5.3

(b) Implement the gradient descent SVM algorithm using Python.

Output:

```
0.835833333333
0.5
0.001
```