CS246: Mining Massive Data Sets

Assignment number: 3 ____

Fill in and include this cover sheet with each of your assignments. It is an honor code violation to write down the wrong time. Assignments and code are due at 5:00 PM on Scoryst and SNAP respectively. Failure to include the coversheet with you assignment will be penalized by 2 points. Each student will have a total of *two* free late periods. *One late period expires at the start of each class.* (Assignments are due on Thursdays, which means the first late period expires on the following Tuesday at 5:00 PM.) Once these late periods are exhausted, any assignments turned in late will be penalized 50% per late period. However, no assignment will be accepted more than one late period after its due date. (If an assignment is due to Thursday then we will not accept it after the following Thursday.)

Your name:Erli Zhou ____

Email:erlizhou@stanford.edu __________________________

SUNet ID:erlizhou ____

Collaborators: ____

I acknowledge and accept the Honor Code.

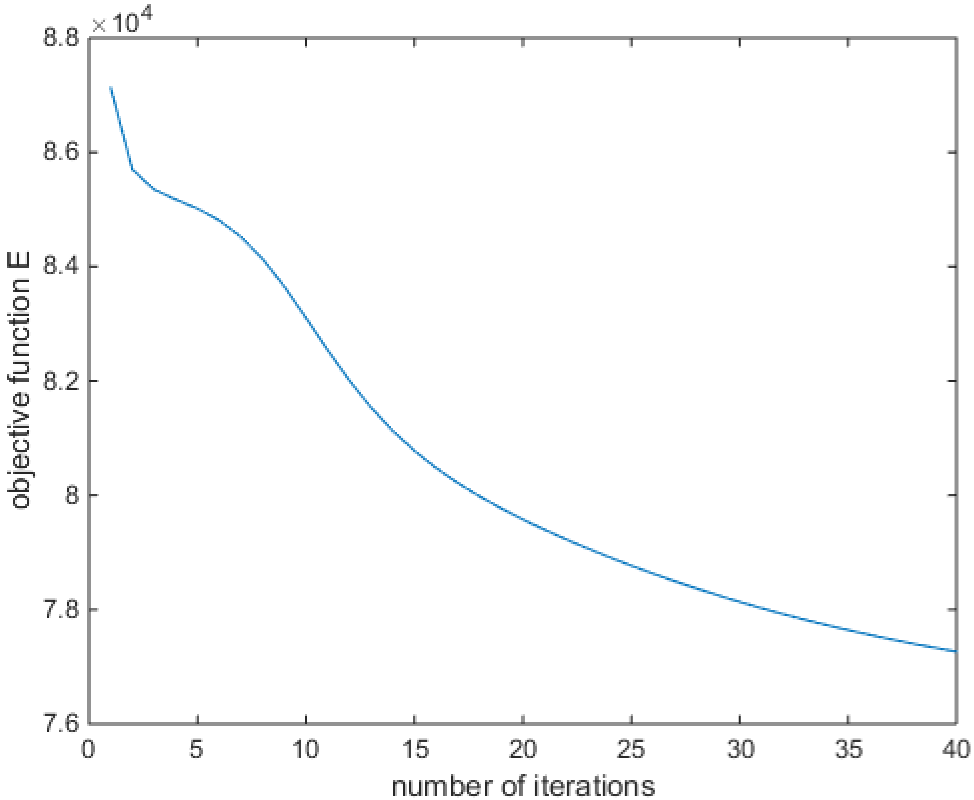
*(Signed)*Erli Zhou____

Answer to Question 1a

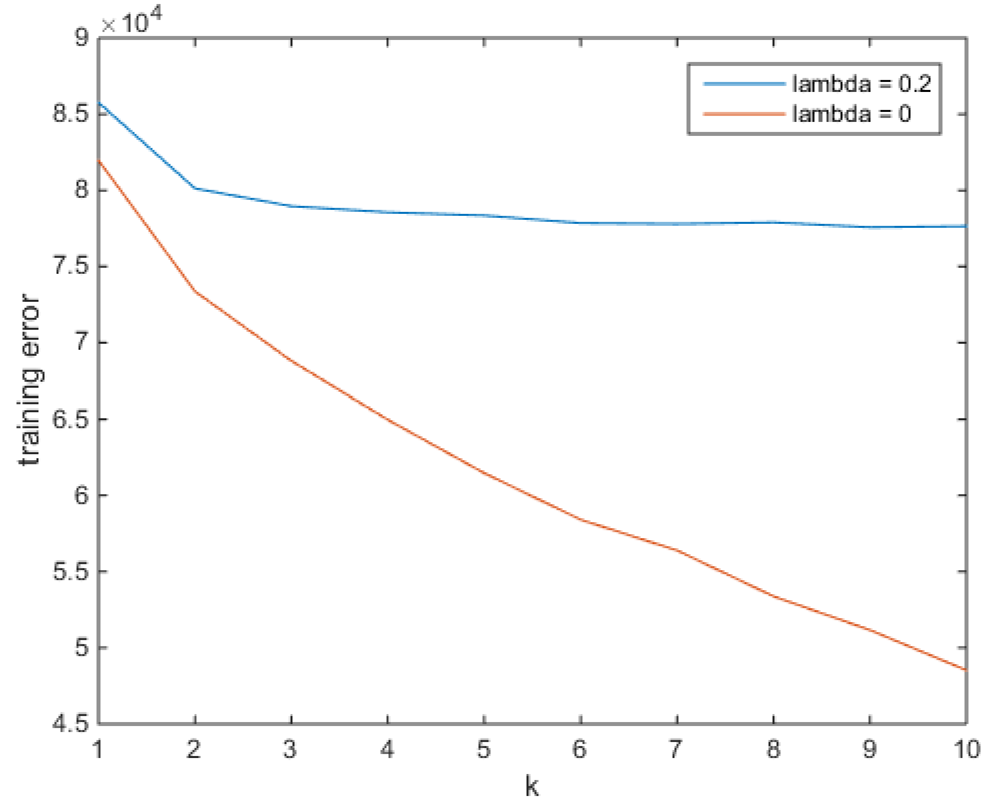
Take derivative of the error function, $\varepsilon_{iu} = 2(R_{iu} - q_i * p_u^T)$
Ignore the constant, $\varepsilon_{iu} = R_{iu} - q_i * p_u^T$
Take derivatives of all constants,
$q_i \leftarrow q_i + \eta * (\varepsilon_{iu} * p_u - \lambda * q_i)$
$p_u \leftarrow p_u + \eta * (\varepsilon_{iu} * q_i - \lambda * p_u)$

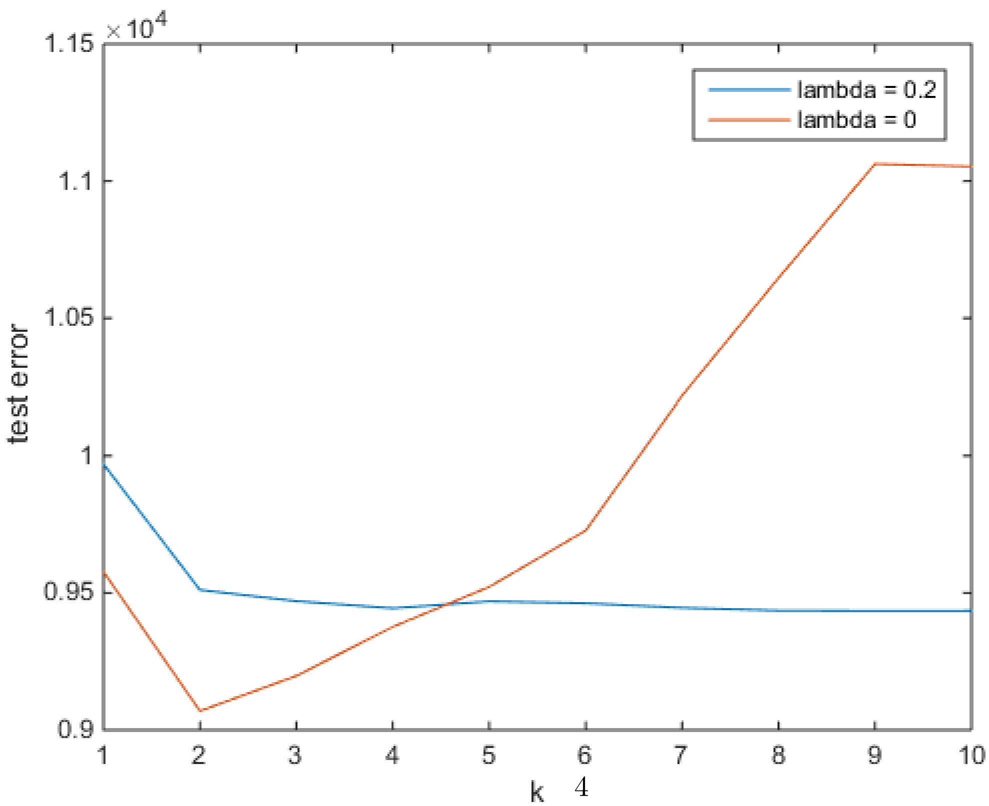
Answer to Question 1b

$\eta = 0.025$



Answer to Question 1c



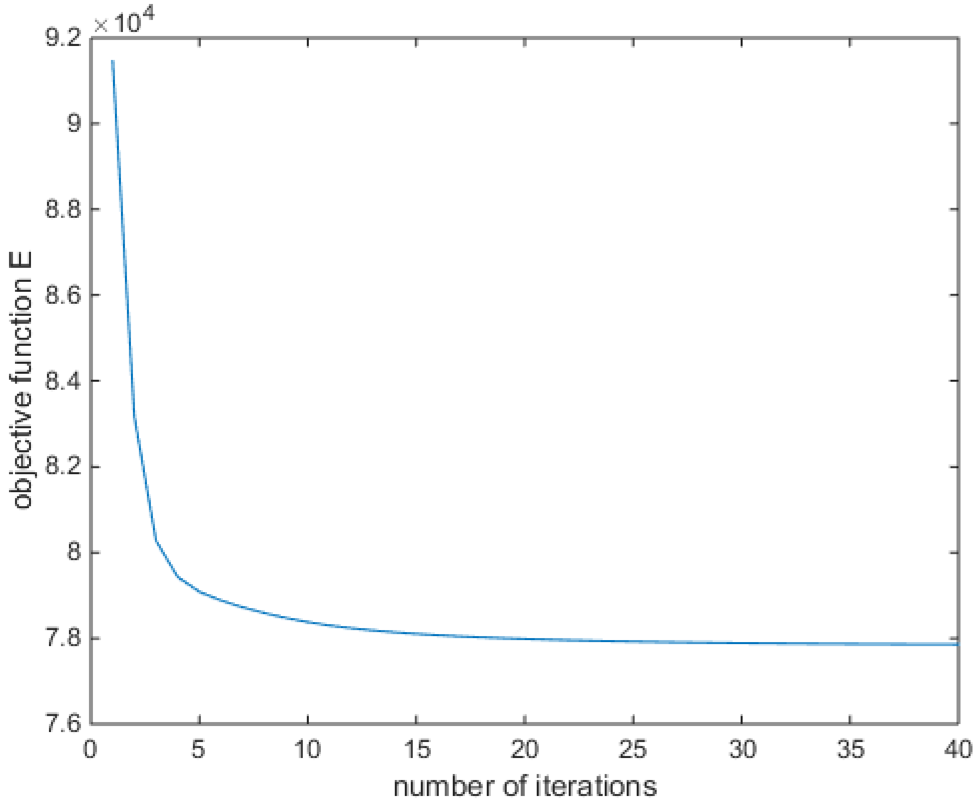


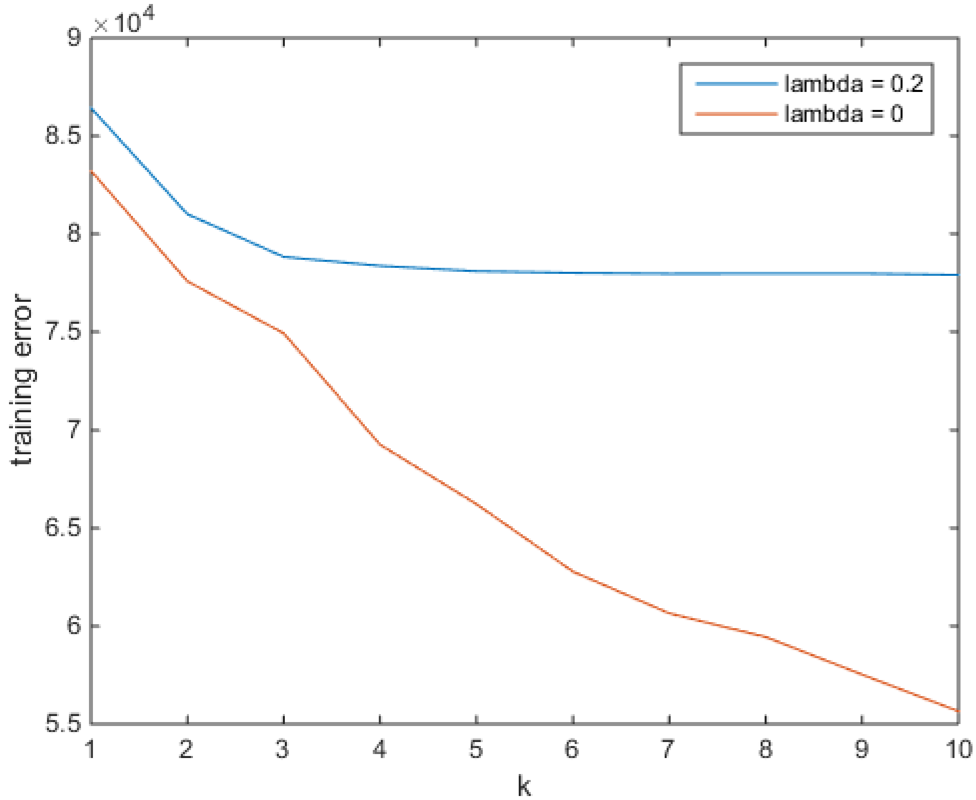
Based on the output, statements B, D and H are valid.

Answer to Question 1d

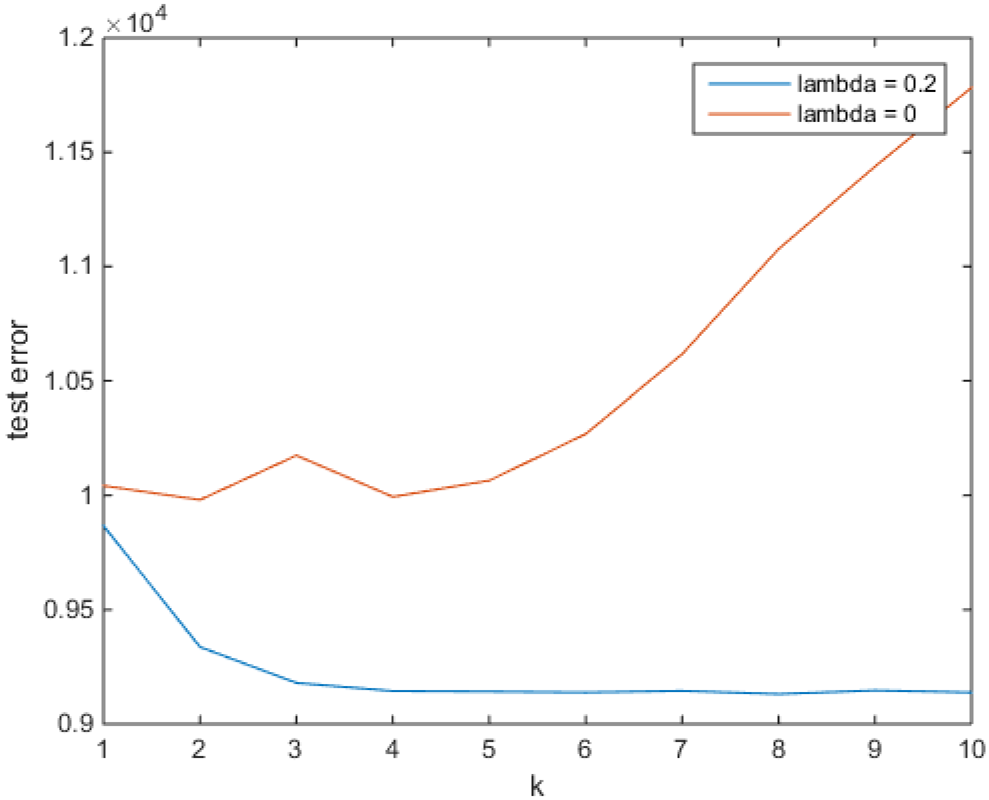
Take derivative of the error function, $\varepsilon_{iu} = 2[R_{iu} - (\mu + b_x + b_i + q_i * p_u^T)]$
Ignore the constant,$\varepsilon_{iu} = R_{iu} - (\mu + b_x + b_i + q_i * p_u^T)$
Take derivatives of all constants,
$q_i \leftarrow q_i + \eta * (\varepsilon_{iu} * p_u - \lambda * q_i)$
$p_u \leftarrow p_u + \eta * (\varepsilon_{iu} * q_i - \lambda * p_u)$
$b_i \leftarrow b_i + \eta * (q_i * p_u^T - \lambda * b_i)$
$p_x \leftarrow b_x + \eta * (q_i * p_u^T - \lambda * b_x)$

η_1 for p and q is 0.1, η_2 for bias parameters is 0.01





×10^4
9
8.5
8
7.5
7
6.5
6
5.5
training error
1
2
3
4
5
6
7
8
9
10
k
lambda = 0.2
lambda = 0



Based on the output, statements B, D and H are valid.

Answer to Question 2a

Based on the definition, $r^{(0)} = \frac{1}{n}1, r^{(k)} = \frac{1-\beta}{n}1 + \beta M r^{(k-1)}$

Also, $r = \frac{1-\beta}{n}1 + \beta M r$

When $k = 0, ||r - r^{(k)}||_1 = ||r - r^{(0)}||_1 \leqslant ||r||_1 + ||r^{(0)}||_1 = 1 + 1 = 2 = 2\beta^{(k)}$, since $||Mr^{(0)}|| = \sum_{i=1}^{n} |\sum_{j=1}^{n} M_{ij} r_j^{(0)}| \leqslant \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}|r_j^{(0)}| = \sum_{j=1}^{n}|r_j^{(0)}|\sum_{i=1}^{n} M_{ij} = ||r^{(0)}||_1$, so after multiple iterations $||r||_1 \leqslant ||r^{(0)}||_1 = 1$

Suppose the equation holds when $k = m, ||r - r^{(m)}||_1 \leqslant 2\beta^m$

For $k = m+1, ||r - r^{(m+1)}||_1 = ||\frac{1-\beta}{n}1 + \beta M r - (\frac{1-\beta}{n}1 + \beta M r^{(m)})||_1 = ||\beta M r - \beta M r^{(m)}||_1 = \beta||M(r - r^{(m)})||_1 \leqslant \beta||r - r^{(m)}||_1 = \beta * 2\beta^m = 2\beta^{m+1}$, thus the equation holds when $k = m+1$ as well.

Thus, we have proved $||r - r^{(k)}||_1 \leqslant 2\beta^k$

Answer to Question 2b

Since $||r - r^{(k)}||_1 \leqslant 2\beta^k < \delta$,

$$k < log_{\beta}\frac{\delta}{2} = -log_{\frac{1}{\beta}}\frac{\delta}{2} = -\frac{1}{log_{\frac{\delta}{2}}\frac{1}{\beta}} = \frac{1}{log_{\frac{2}{\delta}}\frac{1}{\beta}}$$

Since δ is a constant, $K = O(\frac{1}{log(\frac{1}{\beta})})$

Besides, each iteration takes O(m) computing time, so the running time of the power iteration is $O(\frac{m}{log(\frac{1}{\beta})})$

Answer to Question 2c

Assuming taking each step in a random walk takes a unit amount of time,

$T(n) = \beta T(n) + O(1), T(n) = O(\frac{1}{1-\beta})$

Since the total number of random walks are nR, the expected running time is $O(\frac{nR}{1-\beta})$

Answer to Question 2d

The running time of 40 power iterations was 0.004 seconds. It was probably faster than it should be due to the fact that I used numpy package for matrix multiplication.

The running time when R = 1 was 0.0017 seconds.
The running time when R = 3 was 0.0053 seconds.
The running time when R = 5 was 0.0089 seconds.

Average absolute error for nodes:

	R = 1	R = 3	R = 5
K = 10	0.0098	0.0089	0.0087
K = 30	0.0062	0.0053	0.0051
K = 50	0.0049	0.0039	0.0037
K = 100	0.0035	0.0026	0.0024

Answer to Question 3a

After step 1,
s_A(camera, phone) $= \frac{0.8}{3*2} * \sum_{i=1}^{3} \sum_{j=1}^{2} s_B(O_i(1), O_j(2)) = \frac{0.8}{3*2}*(s_B$(nokia, nokia)$+s_B$(nokia, apple)
$+s_B$(kodak, nokia)$+s_B$(kodak, apple)$+s_B$(canon, nokia)$+s_B$(canon, apple))$=s_A$(phone, camera)$=\frac{2}{15}$
s_A(camera, printer)$=s_A$(printer, camera) $= 0$
s_A(phone, printer)$=s_A$(printer, phone) $= 0$
s_A(camera, camera)$=s_A$(printer, printer) $= s_A$(phone, phone) $= 1$
s_B(nokia, hp)$=s_B$(hp, nokia) $= 0$
s_B(nokia, apple)$=s_B$(apple, nokia) $= \frac{2}{5}$
s_B(nokia, kodak)$=s_B$(kodak, nokia) $= \frac{2}{5}$
s_B(nokia, canon)$=s_B$(canon, nokia) $= \frac{2}{5}$
s_B(kodak, apple)$=s_B$(apple, kodak) $= 0$
s_B(kodak, hp)$=s_B$(hp, kodak) $= 0$
s_B(kodak, canon)$=s_B$(canon, kodak) $= \frac{4}{5}$
s_B(canon, apple)$=s_B$(apple, canon) $= 0$
s_B(canon, hp)$=s_B$(hp, canon) $= 0$
s_B(apple, hp)$=s_B$(hp, apple) $= 0$
s_B(apple, apple)$=s_B$(nokia, nokia) $= s_B$(canon, canon)$=s_B$(hp, hp)$=s_B$(kodak, kodak)$=1$

After step 2,
s_A(camera, phone) $= s_A$(phone, camera)$=\frac{22}{75}$
s_A(camera, printer)$=s_A$(printer, camera) $= 0$
s_A(phone, printer)$=s_A$(printer, phone) $= 0$
s_A(camera, camera)$=s_A$(printer, printer) $= s_A$(phone, phone) $= 1$
s_B(nokia, hp)$=s_B$(hp, nokia) $= 0$
s_B(nokia, apple)$=s_B$(apple, nokia) $= \frac{34}{75}$
s_B(nokia, kodak)$=s_B$(kodak, nokia) $= \frac{34}{75}$
s_B(nokia, canon)$=s_B$(canon, nokia) $= \frac{34}{75}$
s_B(kodak, apple)$=s_B$(apple, kodak) $= \frac{8}{75}$
s_B(kodak, hp)$=s_B$(hp, kodak) $= 0$
s_B(kodak, canon)$=s_B$(canon, kodak) $= \frac{4}{5}$
s_B(canon, apple)$=s_B$(apple, canon) $= \frac{34}{75}$
s_B(canon, hp)$=s_B$(hp, canon) $= 0$
s_B(apple, hp)$=s_B$(hp, apple) $= 0$
s_B(apple, apple)$=s_B$(nokia, nokia) $= s_B$(canon, canon)$=s_B$(hp, hp)$=s_B$(kodak, kodak)$=1$

After step 3,
s_A(camera, phone)≈ 0.343
s_A(camera, printer)$=0$

Answer to Question 3b

$$s_A(X,Y) = \frac{C_1}{\displaystyle\sum_{i=1}^{|O(X)|}\sum_{j=1}^{|O(Y)|} W_{X,O_i(X)} W_{Y,O_j(Y)}} \sum_{i=1}^{|O(X)|}\sum_{j=1}^{|O(Y)|} s_B(O_i(X), O_j(Y))$$

$$s_B(x,y) = \frac{C_2}{\displaystyle\sum_{i=1}^{|I(x)|}\sum_{j=1}^{|I(y)|} W_{x,I_i(x)} W_{y,I_j(y)}} \sum_{i=1}^{|I(x)|}\sum_{j=1}^{|I(y)|} s_B(I_i(x), I_j(y))$$

Answer to Question 3c

For $k_{2,1}, A = \{a_1, a_2\}, B = \{b_1\}$
Note $s_B(b_1, b_1)$ always equals one.
Thus, after all 3 steps,
$s_A(a_1, a_2) = \frac{0.8}{1*1} * \sum_{i=1}^{1}\sum_{j=1}^{1} s_B(O_i(1), O_j(2)) = \frac{0.8}{1*1} * (s_B(b_1, b_1)) = s_A(a_2, a_1) = \frac{4}{5}$
$s_A(a_1, a_1) = s_A(a_2, a_2) = 1$

For $k_{2,2}, A = \{a_1, a_2\}, B = \{b_1, b_2\}$
After step 1,
$s_A(a_1, a_2) = \frac{0.8}{2*2} * \sum_{i=1}^{2}\sum_{j=1}^{2} s_B(O_i(1), O_j(2)) = \frac{0.8}{2*2} * (s_B(b_1, b_1) + s_B(b_1, b_2) + s_B(b_2, b_2)) = s_A(a_2, a_1) = \frac{2}{5}$
$s_B(b_1, b_2) = s_B(b_2, b_1) = \frac{2}{5}$
$s_A(a_1, a_1) = s_A(a_2, a_2) = s_B(b_1, b_1) = s_B(b_2, b_2) = 1$

After step 2,
$s_A(a_1, a_2) = s_A(a_2, a_1) = \frac{14}{25}$
$s_B(b_1, b_2) = s_B(b_2, b_1) = \frac{14}{25}$
$s_A(a_1, a_1) = s_A(a_2, a_2) = s_B(b_1, b_1) = s_B(b_2, b_2) = 1$

After step 3,
$s_A(a_1, a_2) = s_A(a_2, a_1) = \frac{78}{125}$
$s_A(a_1, a_1) = s_A(a_2, a_2) = 1$

Answer to Question 4a

i. According to graph theory, $\rho(S) = \frac{1}{2|S|} \sum_{v \in S} deg_s(v)$

So $2|E(S)| = \sum_{v \in S} deg_s(v) \geqslant \displaystyle\sum_{v \in S \setminus A(s)} deg_s(v) \geqslant (|S| - |A(S)|) * 2(1+\varepsilon)\frac{|E(S)|}{|S|}$

Thus we get $|S| \geqslant (|S| - |A(S)|)(1+\varepsilon)$, which leads to $(1+\varepsilon) \geqslant \varepsilon|S|$

As a result, $|A(S)| \geqslant \frac{\varepsilon}{1+\varepsilon}|S|$

ii. Suppose before each iteration we have n nodes remaining. After the ith iteration we are left with at most $n - \dfrac{\varepsilon}{1+\varepsilon}n = \dfrac{1}{1+\varepsilon}n$ nodes. So the factor of decrease is at least $1+\varepsilon$. Thus, the algorithm terminates in at most $log_{1+\varepsilon}(n)$ iterations.

Answer to Question 4b

i. Assume for certain $v \in S^*$, we have: $deg_{S^*}(v) < \rho^*(G) = \rho(S^*)$

Then $\rho(S^*\backslash v) = \dfrac{|E[S^*]| - deg_{S^*}(v)}{|S^*| - 1} = \rho(S^*) + \dfrac{\rho(S^*) - deg_{S^*}(v)}{|S^*| - 1} > \rho(S^*)$

This contradicts the assumption that S^* is the densest subgraph of G.

Using proof by contradiction, we get for any $v \in S^*$, we have: $deg_{S^*}(v) \geqslant \rho^*(G)$

ii. Since $v \in A(S), deg_{S^*}(v) \geqslant 2(1+\varepsilon)\rho(S)$

Because this is the first iteration such that $S^* \bigcap A(S) \neq \varnothing$, we have $S^* \subseteq S$ and $deg_S(v) \geq deg_{S^*}(v)$

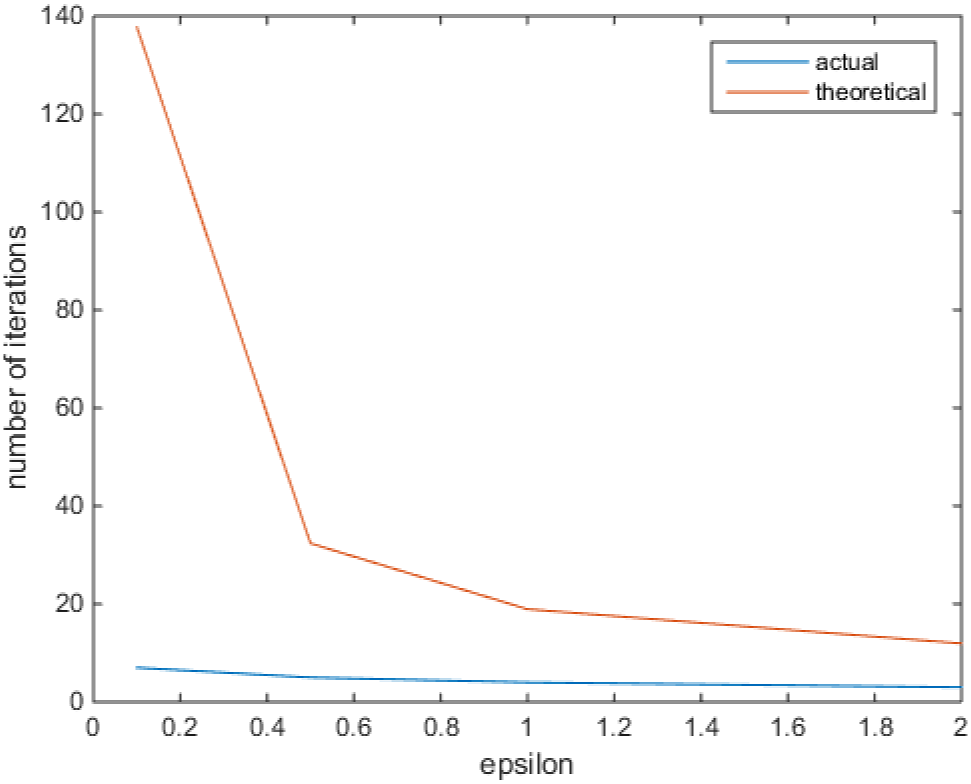
From this we can deduct $\rho^*(G) \leqslant deg_{S^*}(v) \leqslant deg_S(v) \leqslant 2(1+\varepsilon)\rho(S)$

iii. We know from ii. that $\exists S$ s.t. $2(1+\varepsilon)\rho(S) \geqslant \rho^*(G)$, because $S^* \subseteq V$, we start with S = V and remove vertexes until $S = \varnothing$

As a result, $\tilde{S}$ is the set which maximizes $\rho(S)$, thus $\rho(S) \geqslant \dfrac{1}{2(1+\varepsilon)}\rho^*(G)$

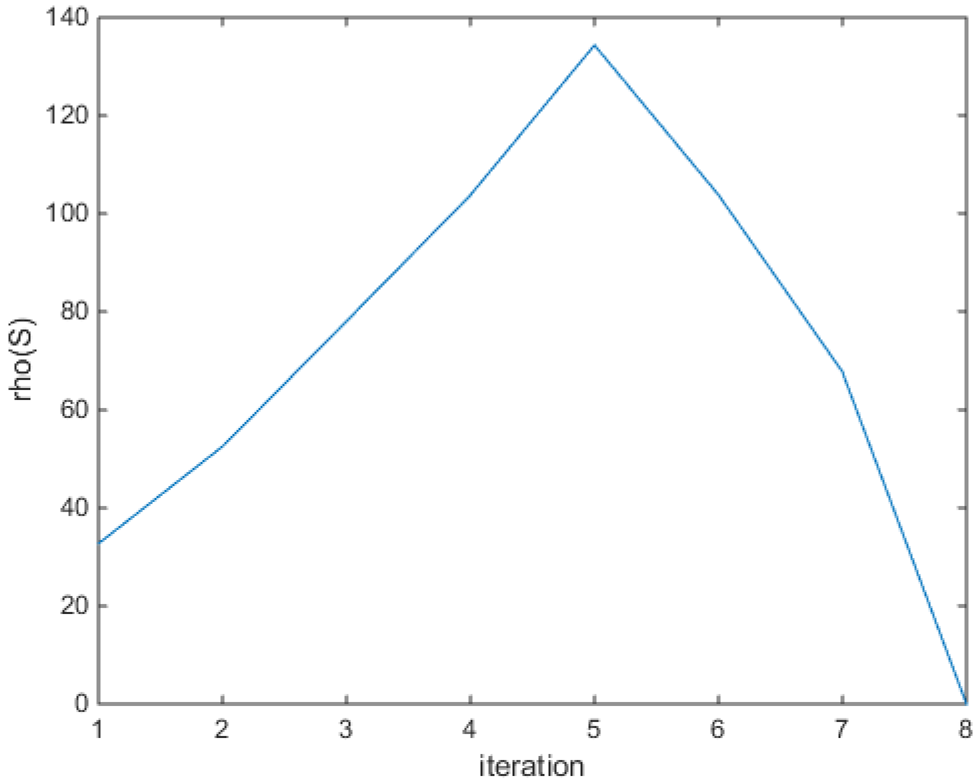
Answer to Question 4c

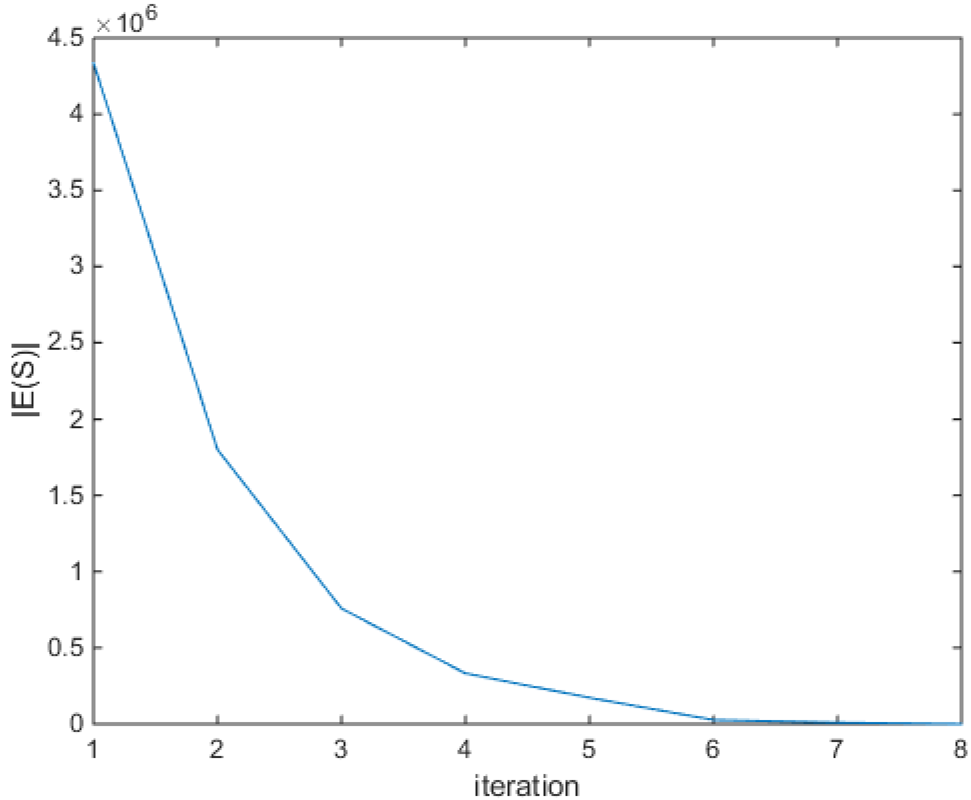
i.

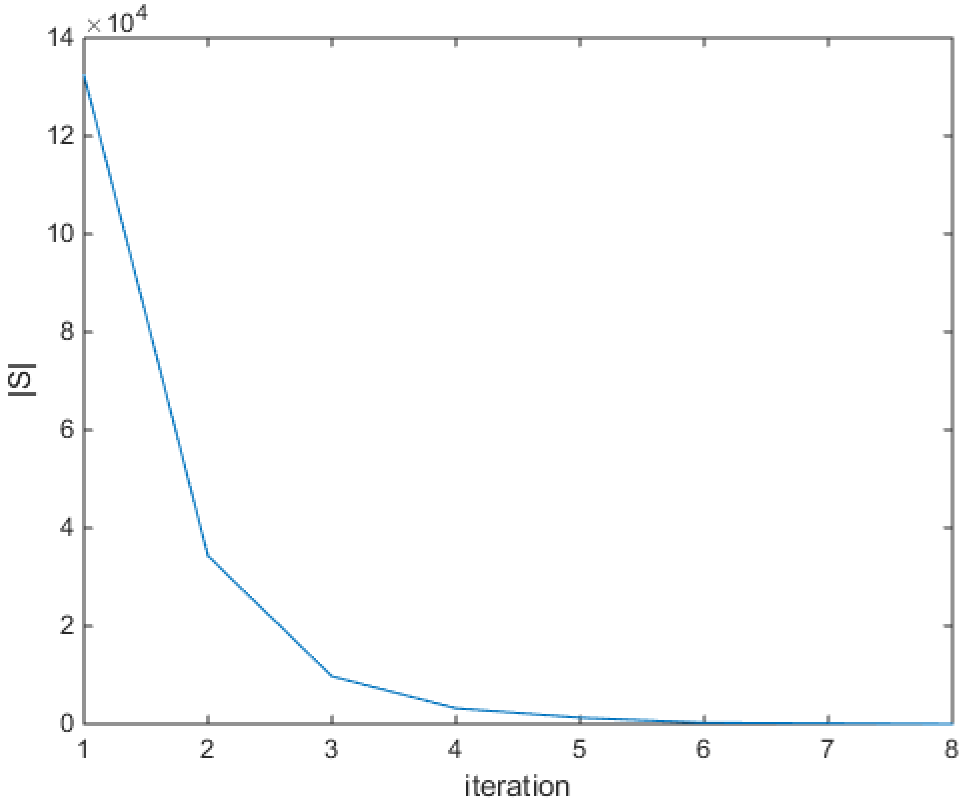


The actual data set takes much fewer iterations than the theoretical bounds.

ii.

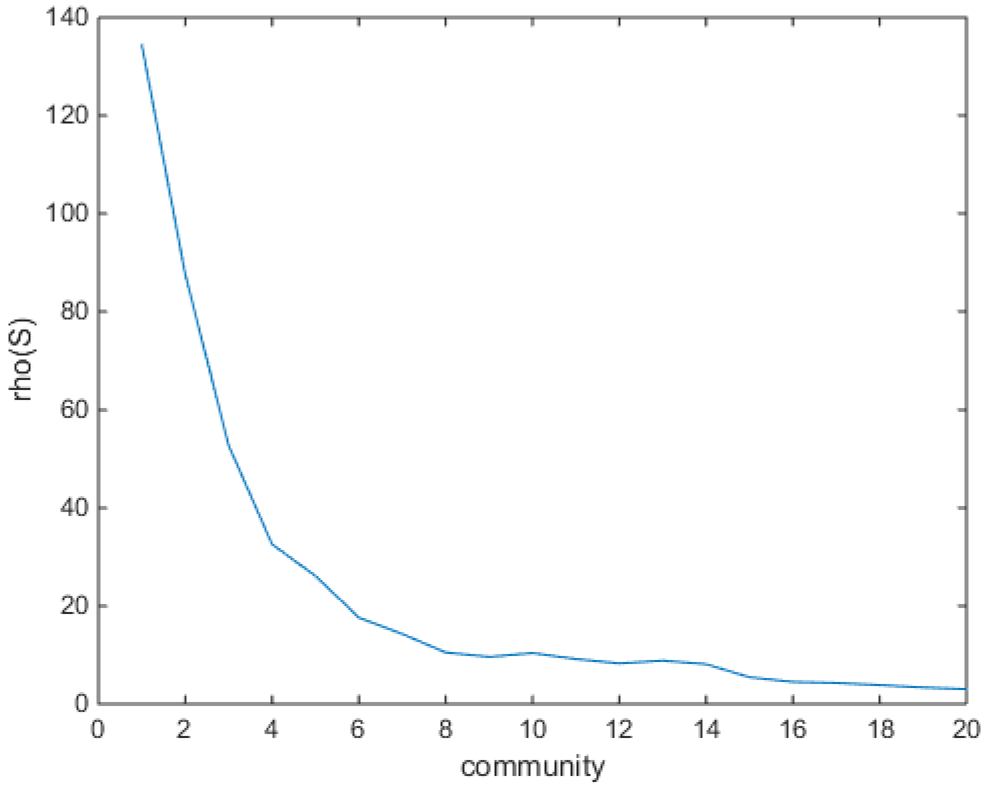


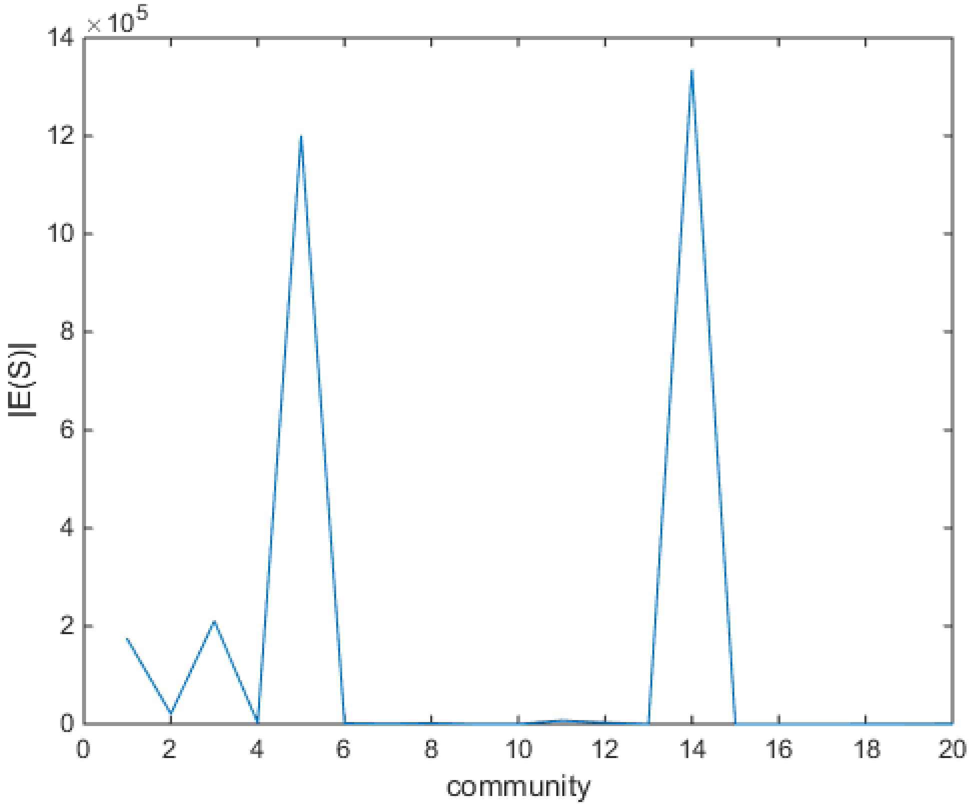

×10^6
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
|E(S)|
1
2
3
4
5
6
7
8
iteration



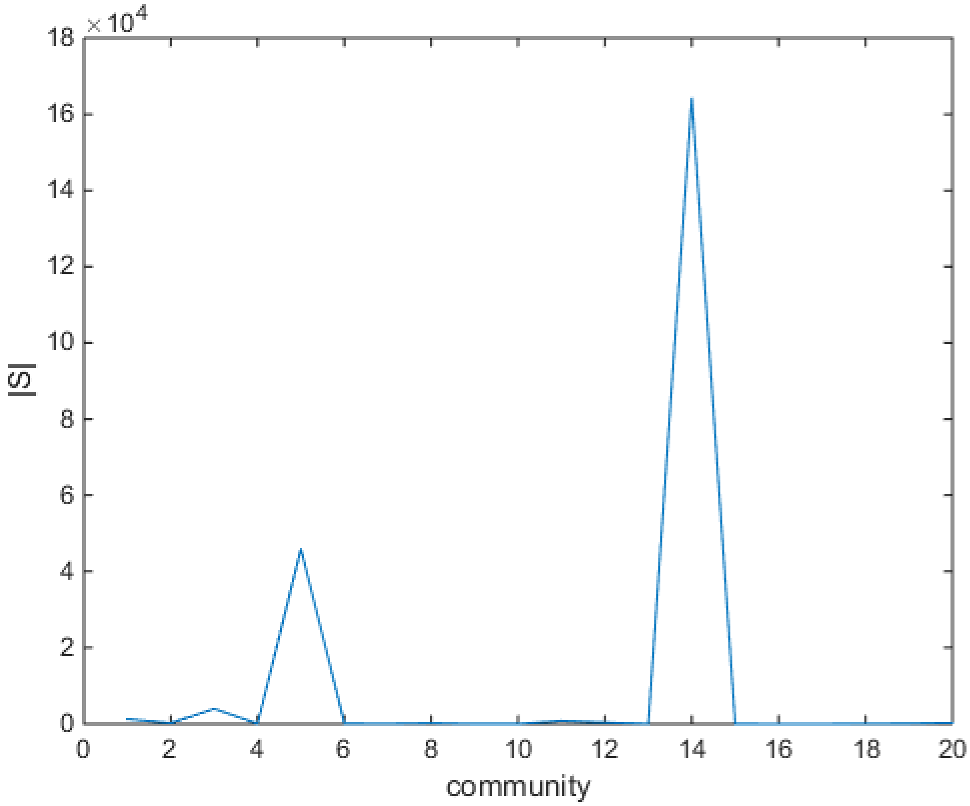
×10^4
14
12
10
8
6
4
2
0
|S|
1
2
3
4
5
6
7
8
iteration

iii.





×10^5
14
12
10
8
6
4
2
0
|E(S)|
0
2
4
6
8
10
12
14
16
18
20
community



×10^4
18
16
14
12
10
8
6
4
2
0
|S|
0
2
4
6
8
10
12
14
16
18
20
community