Written Assignment 3 solution
SENG 474/CSC 578D

April 15, 2019

Question 1

With transition matrix, $\mathbf{M} = \begin{bmatrix} 1/3 & 1/2 & 0 \\ 1/3 & 0 & 1/2 \\ 1/3 & 1/2 & 1/2 \end{bmatrix}$;

intial probability distribution, $\mathbf{v} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$, $\beta = 0.8$; number of pages, n = 3

and $\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, we can can compute page rank by iterating and updating $\mathbf{v}$ by the following formula,

$\mathbf{v} = \beta\mathbf{M}\mathbf{v} + (1-\beta)\mathbf{e}/n$

Updations to $\mathbf{v}$: $\begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} \rightarrow \begin{bmatrix} 0.289 \\ 0.289 \\ 0.422 \end{bmatrix} \rightarrow \begin{bmatrix} 0.259 \\ 0.312 \\ 0.428 \end{bmatrix}$ .... $\rightarrow \begin{bmatrix} 0.259 \\ 0.308 \\ 0.432 \end{bmatrix}$

Question 2

Proof by induction:-

Let $v = \begin{bmatrix} x \\ y \\ y \\ y \end{bmatrix}$ $M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$

For n=0, $M^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$M^0v = \begin{bmatrix} x \\ y \\ y \\ y \end{bmatrix}$$

As the second, third and fourth component are equal in M^0v, the statement holds true for n = 0.

Let's assume that the statement holds true for any specific number k such that the second, third and the fourth components are equal in M^kv and let $M^kv = \begin{bmatrix} a \\ b \\ b \\ b \end{bmatrix}$

Now we need to check if the statement holds true for n=k+1.

$$M^{k+1}v = MM^kv = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ b \\ b \end{bmatrix} = \begin{bmatrix} 1/2a + b \\ 1/3a + 1/2b \\ 1/3a + 1/2b \\ 1/3a + 1/2b \end{bmatrix}$$

As it can be seen that the second, third and the fourth components are equal in $M^{k+1}v$. Thus, the statement holds true.

Question 3

With transition matrix, $\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$;

intial probability distribution, $\mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, teleportation vector, $\mathbf{e} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $\beta = 0.8$ and number of trusted pages,n =1 , we can can compute trust rank by iterating and updating $\mathbf{v}$ by the following formula,

$\mathbf{v} = \beta\mathbf{M}\mathbf{v} + (1-\beta)\mathbf{e}/n$

Updations to $\mathbf{v}$: $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.4 \\ 0.2 \\ 0 \\ 0.4 \end{bmatrix} \rightarrow \begin{bmatrix} 0.08 \\ 0.467 \\ 0.267 \\ 0.187 \end{bmatrix} \begin{bmatrix} 0.08 \\ 0.467 \\ 0.267 \\ 0.187 \end{bmatrix} \text{............} \rightarrow \begin{bmatrix} 0.269 \\ 0.358 \\ 0.158 \\ 0.215 \end{bmatrix}$

Thus, Trustrank of the web pages will be $\begin{bmatrix} 0.269 \\ 0.358 \\ 0.158 \\ 0.215 \end{bmatrix}$

In order to find spammass, we need to find page rank of all the pages. The page rank computation will be similar to the above except the fact that now the number of trusted pages, n = 4 , teleportation vector , $\mathbf{e} = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}$ and initial probability distribution, $\mathbf{v} = \begin{bmatrix}1/4\\1/4\\1/4\\1/4\end{bmatrix}$

Thus we can can compute page rank by iterating and updating $\mathbf{v}$ by the following formula,

$\mathbf{v} = \beta\mathbf{M}\mathbf{v} + (1-\beta)\mathbf{e}/n$

Updations to $\mathbf{v}$: $\begin{bmatrix}0.25\\0.25\\0.25\\0.25\end{bmatrix} \rightarrow \begin{bmatrix}0.35\\0.217\\0.217\\0.217\end{bmatrix} \rightarrow \begin{bmatrix}0.31\\0.23\\0.23\\0.23\end{bmatrix}$ .... $\rightarrow \begin{bmatrix}0.321\\0.226\\0.226\\0.226\end{bmatrix}$

Thus, PageRank of the web pages will be $\begin{bmatrix}0.321\\0.226\\0.226\\0.226\end{bmatrix}$

According to the definition of spammass, if page p has PageRank r and TrustRank t, then the spam mass of p is $(r-t)/r$

$$Spammass - \begin{bmatrix}(0.321-0.269)/0.321\\(0.226-0.358)/0.226)\\(0.226-0.158)/0.226)\\(0.226-0.215)/0.226\end{bmatrix} = \begin{bmatrix}0.162\\-0.584\\0.301\\0.049\end{bmatrix}$$

Question 4

According to the HITS algorithm posted in the lecture slides,

Initialize $\mathbf{a} = \mathbf{h} = \frac{\mathbf{1}}{||\mathbf{1}||}$, where $\mathbf{1} = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}$ and $||\mathbf{1}||$ is the magnitude of vector $\mathbf{1}$ i.e $||\mathbf{1}|| = \sqrt{1+1+1+1} = 2$

For each iteration, do:

- $a = L^T h$
- $a = \frac{a}{||a||}$
- $h = La$
- $h = \frac{h}{||h||}$

where $\mathbf{L} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$

According to the above algorithm **a** and **h** will be updated as follows:

$$\mathbf{a} \rightarrow \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 0.327 \\ 0.545 \\ 0.545 \\ 0.545 \end{bmatrix} \rightarrow \begin{bmatrix} 0.252 \\ 0.573 \\ 0.573 \\ 0.527 \end{bmatrix} \cdots\cdots \rightarrow \begin{bmatrix} 0.174 \\ 0.603 \\ 0.603 \\ 0.491 \end{bmatrix}$$

$$\mathbf{h} \rightarrow \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 0.707 \\ 0.471 \\ 0.235 \\ 0.471 \end{bmatrix} \rightarrow \begin{bmatrix} 0.751 \\ 0.401 \\ 0.150 \\ 0.501 \end{bmatrix} \cdots\cdots \rightarrow \begin{bmatrix} 0.773 \\ 0.303 \\ 0.079 \\ 0.550 \end{bmatrix}$$

Other normalization methods like the one specified in section 5.5.2 in the textbook can be used as well.

Question 5

By picking up the cluster with smallest radius on each iteration, the execution might look something like the figures below. Please note that out of two equidistant clusters, one was picked up randomly. So the execution may be different from the one shown here.

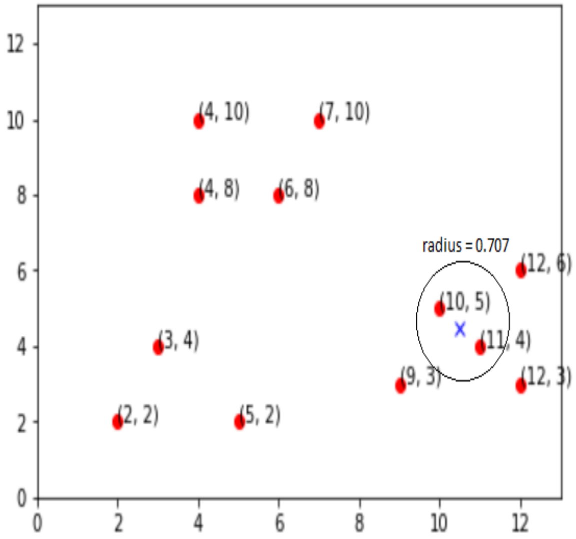

Figure 1:

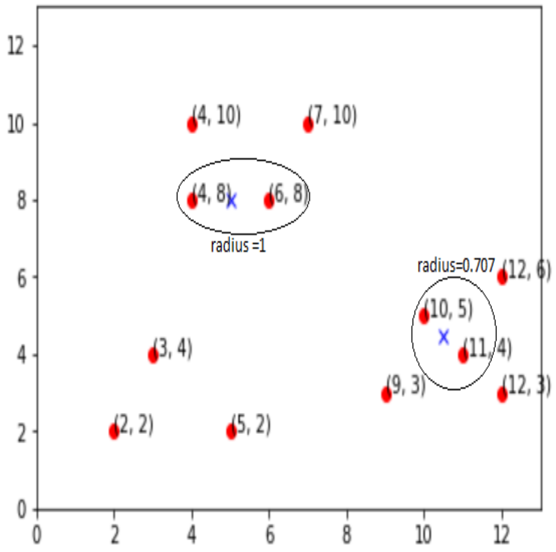

Figure 2:

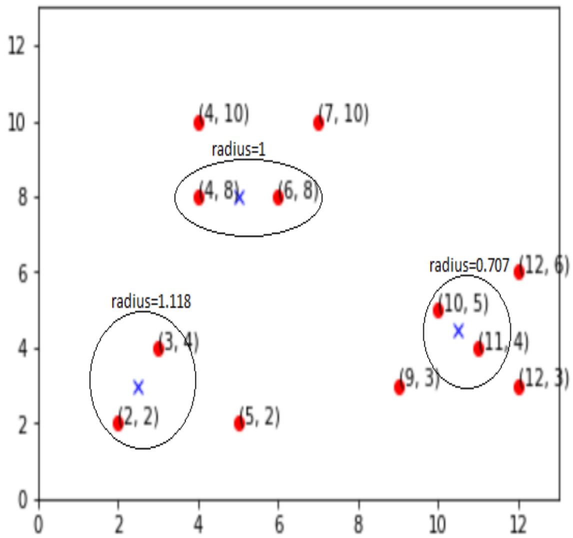

Figure 3:

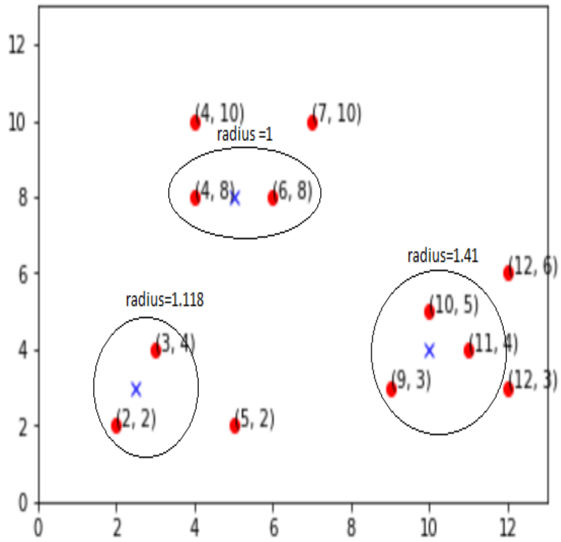

Figure 4:

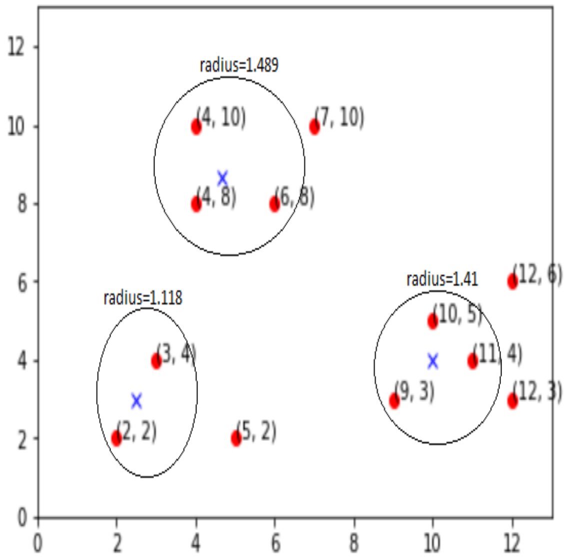

Figure 5:

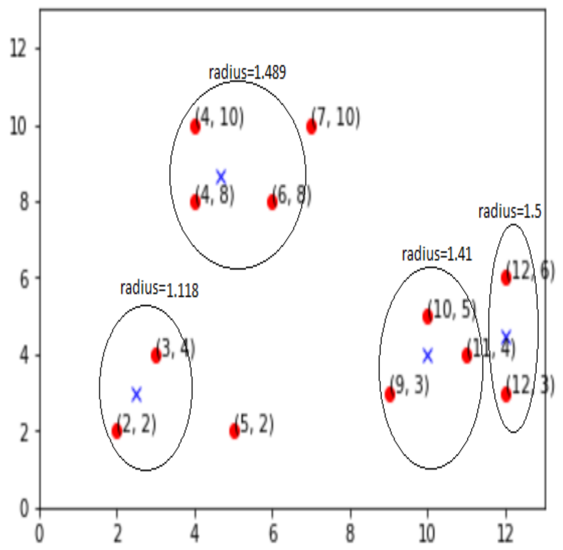

Figure 6:

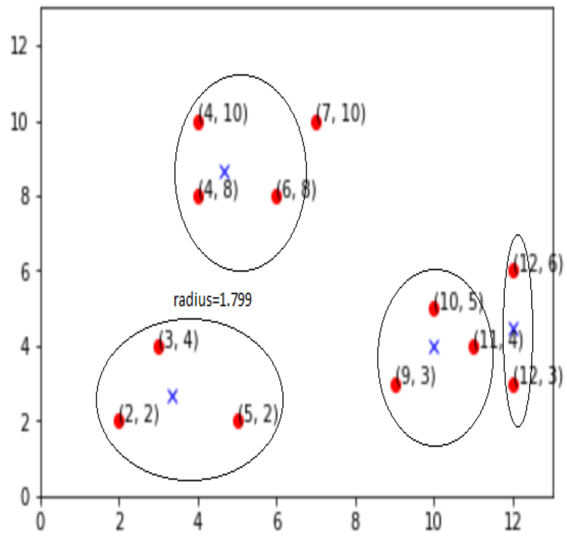

Figure 7:

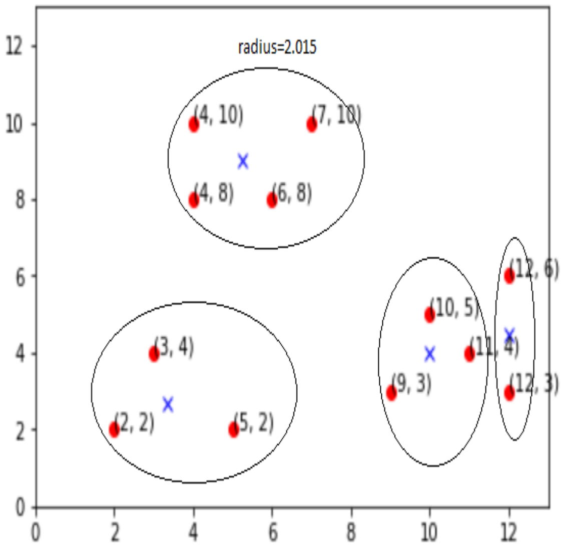

Figure 8:

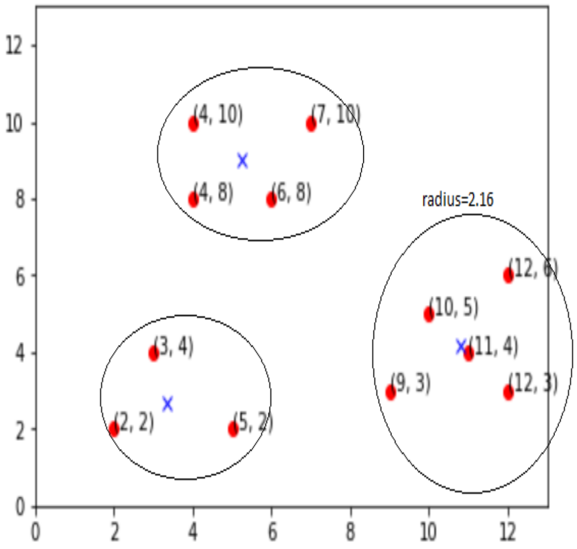

Figure 9:

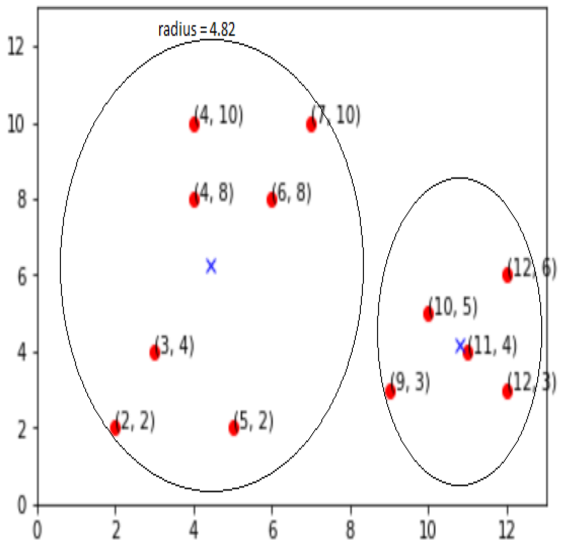

Figure 10:

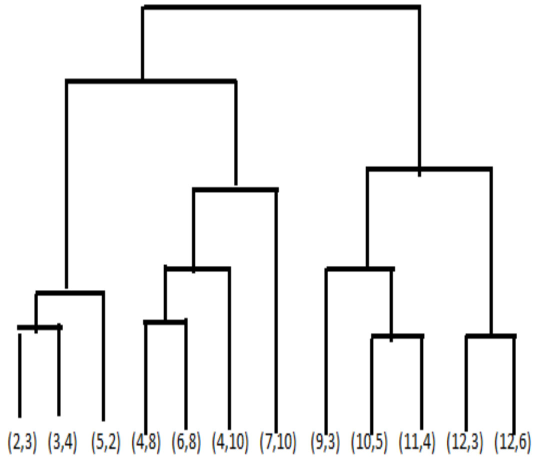

Figure 11:

Question 6

```
WHILE there are fewer than k points DO
Add the point whose minimum distance from the selectedpoints is as large as possible;
END;
```

Following up with the above algorithm, let S be the set of points that we have picked so far.
Then we need to find a point(x,y) which maximizes mindist((x,y),S) ,
where mindist((x,y),S) is the distance from point (x,y) to the closest point in S.

Initially, S=[(3,4)]
By visual inspection, it can be noticed that the point that is farthest from (3,4) can be (4,10), (7,10), (12,6) and (12,3)
$d(3,4),(4,10) = \sqrt{37}$
$d(3,4),(7,10) = \sqrt{52}$
$d(3,4),(12,6) = \sqrt{85}$
$d(3,4),(12,3) = \sqrt{82}$
Since (12,6) is the farthest. We pick it and add it to S.

Now $S = [(3,4),(12,6)]$
By visual inspection, it can be noticed that point that is farthest from both (3,4) and (12,6) can be either (4,10) or (7,10)
$mindist((4,10),S) = min(d((4,10),(3,4)), d((4,10),(12,6))) = min(\sqrt{37}, \sqrt{80}) = \sqrt{37}$
$mindist((7,10),S) = min(d((7,10),(3,4)), d((7,10),(12,6))) = min(\sqrt{42}, \sqrt{41}) = \sqrt{41}$
Since $mindist((7,10),S) > mindist((4,10),S)$, $(7,10)$ will be picked up.
Finally, $S = [(3,4),(12,6),(7,10)]$

Question 7

a.

Since query x can be assigned to any of the advertisor, both x queries will be assigned eventually. After the assignments of x, one of the following cases will occur:

1. The budget of one specific advertisor will be exhausted
2. The budget of two advertisor reduce by 1 each

Since y can be assigned to 2 advertisors, in both of the cases discussed above, there will be at least total budget of 2 remaining for the advertisors that can take in y. So, both y will be assigned eventually. So, at least 4 of these 6 queries will be assigned in any case.

b.

The query can be xxzz
In a specific case, both x will be assigned to C and thus z will remain unassigned.
Since optimal offline algorithm can assign all 4 of these queries and our greedy might assign only 2 out of these queries, the ratio will be $\frac{2}{4}$
Other queries like yyzz can exist too.