Homework 1

github.com/stefgmz

November 26, 2022

Question 1

Let $\mathcal{S}$ be any set of nodes of the graph. We denote by $\mathbb{1}_{\mathcal{S}} \in \mathbb{R}^N$ the vector

$$\mathbb{1}_{\mathcal{S}} = \begin{cases} 1, \text{ if } i \in \mathcal{S} \\ 0, \text{ else.} \end{cases}$$

Let $G_{\mathcal{S}}$ be the matrix defined by

$$G_{\mathcal{S},ij} = \begin{cases} \beta M_{ij} + (1-\beta)/|\mathcal{S}|, \text{ if } i \in \mathcal{S} \\ \beta M_{ij}, \text{ else} \end{cases}$$

with $M_{ij} = 1/d_i$ if there is an edge from i to j and else 0. Here d_i denotes the outdegree of node i. If M is (column) stochastic, then so is $G_{\mathcal{S}}$, i.e. there exists an eigenvector r with eigenvalue 1 and $\sum_{i=1}^N r_i = 1$. This is the personalized PageRank vector and we have

$$G_{\mathcal{S}} v = v.$$

We can write $G_{\mathcal{S}} = \beta M + (1-\beta)/|\mathcal{S}| \mathbb{1}_{\mathcal{S}} \mathbb{1}^\top$. Then

$$\begin{aligned} v = G_{\mathcal{S}} v &= \beta M v + \frac{1-\beta}{|\mathcal{S}|} \mathbb{1}_{\mathcal{S}} \mathbb{1}^\top v \\ &= \beta M v + \frac{1-\beta}{|\mathcal{S}|} \mathbb{1}_{\mathcal{S}} (\mathbb{1}^\top v) \\ &= \beta M v + \frac{1-\beta}{|\mathcal{S}|} \mathbb{1}_{\mathcal{S}} \left(\sum_{i=1}^N v_i \right) \\ &= \beta M v + \frac{1-\beta}{|\mathcal{S}|} \mathbb{1}_{\mathcal{S}} \end{aligned} \tag{1}$$

since $\sum_{i=1}^N v_i = 1$.

Assume we are given multiple sets of teleport vectors $\mathcal{S}, \mathcal{S}_1, \ldots, \mathcal{S}_m$ such that

$$\mathbb{1}_{\mathcal{S}} = \lambda_1 \mathbb{1}_{\mathcal{S}_1} + \ldots + \lambda_m \mathbb{1}_{\mathcal{S}_m}$$

for some integers $\lambda_1, \ldots, \lambda_m$. Further we assume that $v_{\mathcal{S}_i}$ is the PageRank vector for the teleport set $\mathcal{S}_i$. We claim that

$$v_{\mathcal{S}} = \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 v_{\mathcal{S}_1} + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m v_{\mathcal{S}_m}.$$

is the PageRank vector for the set $\mathcal{S}$. From (1) we obtain

$$\begin{aligned}
v_{\mathcal{S}} &= \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 v_{\mathcal{S}_1} + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m v_{\mathcal{S}_m} \\
&= \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 \left(\beta M v_{\mathcal{S}_1} + \frac{1-\beta}{|\mathcal{S}_1|}\mathbb{1}_{\mathcal{S}_1}\right) + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m \left(\beta M v_{\mathcal{S}_m} + \frac{1-\beta}{|\mathcal{S}_m|}\mathbb{1}_{\mathcal{S}_m}\right) \\
&= \beta M \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 v_{\mathcal{S}_1} + \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 \frac{1-\beta}{|\mathcal{S}_1|}\mathbb{1}_{\mathcal{S}_1} + \ldots + \beta M \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m v_{\mathcal{S}_m} + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m \frac{1-\beta}{|\mathcal{S}_m|}\mathbb{1}_{\mathcal{S}_m} \\
&= \beta M \left(\frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 v_{\mathcal{S}_1} + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m v_{\mathcal{S}_m}\right) + \frac{1-\beta}{|\mathcal{S}|}\left(\lambda_1 \mathbb{1}_{\mathcal{S}_1} + \ldots + \lambda_m \mathbb{1}_{\mathcal{S}_m}\right) \\
&= \beta M v_{\mathcal{S}} + \frac{1-\beta}{|\mathcal{S}|}\mathbb{1}_{\mathcal{S}}
\end{aligned}$$

so that indeed $v_{\mathcal{S}}$ is an eigenvector to the eigenvalue 1 of $G_{\mathcal{S}}$. It remains to show that it is normalized:

$$\begin{aligned}
\sum_{i=1}^{N} v_{\mathcal{S},i} &= \sum_{i=1}^{N}\left(\frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 v_{\mathcal{S}_1,i} + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m v_{\mathcal{S}_m,i}\right) \\
&= \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 \sum_{i=1}^{N} v_{\mathcal{S}_1,i} + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m \sum_{i=1}^{N} v_{\mathcal{S}_m,i} \\
&= \frac{|\mathcal{S}_1|}{|\mathcal{S}|}\lambda_1 + \ldots + \frac{|\mathcal{S}_m|}{|\mathcal{S}|}\lambda_m \\
&= |\mathcal{S}|^{-1}\left(|\mathcal{S}_1|\lambda_1 + \ldots + |\mathcal{S}_m|\lambda_m\right) \\
&= |\mathcal{S}|^{-1}\left(\lambda_1 \mathbb{1}^\top \mathbb{1}_{\mathcal{S}_1} + \ldots + \lambda_m \mathbb{1}^\top \mathbb{1}_{\mathcal{S}_m}\right) \\
&= |\mathcal{S}|^{-1}\mathbb{1}^\top\left(\lambda_1 \mathbb{1}_{\mathcal{S}_1} + \ldots + \lambda_m \mathbb{1}_{\mathcal{S}_m}\right) \\
&= |\mathcal{S}|^{-1}\mathbb{1}^\top \mathbb{1}_{\mathcal{S}} \\
&= |\mathcal{S}|^{-1}|\mathcal{S}| \\
&= 1.
\end{aligned}$$

Question 1.1

By abuse of notation we will denote by $A, B, \ldots$ the teleport sets of user $A, B, \ldots$. Let E denote the teleport set of Eloise. Then $\mathbb{1}_E = \mathbb{1}_A - \mathbb{1}_B + \mathbb{1}_C - 2 \cdot \mathbb{1}_D$. By our previous thoughts the PageRank vector for Eloise can be computed by

$$v_E = 3v_A - 3v_B + 3v_C - 2v_D.$$

Question 1.2

We cannot compute the PageRank vector for Felicity. We would need another user with either 4 or 5 in the teleport set. With the current data we have the canonical teleport sets $\{1\},\{2\},\{3\},\{4,5\}$.

Question 1.3

The matrix for user Glynnis is given by

$$G_G = \beta M + \frac{1-\beta}{10}\begin{pmatrix}1\\2\\3\\2\\2\end{pmatrix}.$$

Looking at the canonical teleport sets we set

$$v_G = \frac{9}{10}v_A + \frac{6}{10}v_C - \frac{1}{10}v_E.$$

Then

$$\begin{aligned}
v_G &= \frac{9}{10}v_A + \frac{6}{10}v_C - \frac{4}{10}v_D - \frac{1}{10}v_E \\
&= \frac{9}{10}\left(\beta M v_A + \frac{1-\beta}{3}\mathbb{1}_A\right) + \frac{6}{10}\left(\beta M v_C + \frac{1-\beta}{3}\mathbb{1}_C\right) \\
&\quad - \frac{4}{10}\left(\beta M v_D + (1-\beta)\mathbb{1}_D\right) - \frac{1}{10}\left(\beta M v_E + (1-\beta)\mathbb{1}_E\right) \\
&= \beta M\left(\frac{9}{10}v_A + \frac{6}{10}v_C - \frac{4}{10}v_D - \frac{1}{10}v_E\right) + \frac{1-\beta}{10}\left(3\cdot\mathbb{1}_A + 2\cdot\mathbb{1}_C - 4\cdot\mathbb{1}_D - \mathbb{1}_E\right) \\
&= G_G v_G
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i=1}^N v_{G,i} &= \sum_{i=1}^N \left(\frac{9}{10}v_{A,i} + \frac{6}{10}v_{C,i} - \frac{4}{10}v_{D,i} - \frac{1}{10}v_{E,i}\right) \\
&= \frac{9}{10}\sum_{i=1}^N v_{A,i} + \frac{6}{10}\sum_{i=1}^N v_{C,i} - \frac{4}{10}\sum_{i=1}^N v_{D,i} - \frac{1}{10}\sum_{i=1}^N v_{E,i} \\
&= \frac{9}{10} + \frac{6}{10} - \frac{4}{10} - \frac{1}{10} \\
&= 1
\end{aligned}$$

since every PageRank vector's entries sum up to 1.

Question 1.4

Any vector that satisfies the conditions discussed on page 1 can be computed, i.e. $v = \lambda_1 v_{\mathcal{S}_1} + \ldots + \lambda_m v_{\mathcal{S}_m}$ must be a linear combination of vectors in V satisfying

$$\mathbb{1}_{\mathcal{S}} = \lambda_1 \frac{|\mathcal{S}|}{|\mathcal{S}_1|} \mathbb{1}_{\mathcal{S}_1} + \ldots + \lambda_m \frac{|\mathcal{S}|}{|\mathcal{S}_m|} \mathbb{1}_{\mathcal{S}_m}$$

Question 1.5

This is a special case of the initial discussions on page 1 with $\mathcal{S} = V$.

Question 2

After one iteration:

$P(Y_1 = +) = \frac{1}{2}$
$P(Y_2 = +) = \frac{3}{8}$
$P(Y_3 = +) = \frac{5}{8}$
$P(Y_4 = +) = \frac{15}{32}$
$P(Y_8 = +) = \frac{31}{64}$
$P(Y_9 = +) = \frac{31}{128}$

After two iterations:

$P(Y_1 = +) = \frac{1}{2}$
$P(Y_2 = +) = \frac{51}{128}$
$P(Y_3 = +) = \frac{81}{128}$
$P(Y_4 = +) = \frac{241}{512}$
$P(Y_8 = +) = \frac{365}{1024}$
$P(Y_9 = +) = \frac{365}{2048}$

After three iterations:

$P(Y_1 = +) = \frac{33}{64}$
$P(Y_2 = +) = \frac{829}{2048}$
$P(Y_3 = +) = \frac{1311}{2048}$
$P(Y_4 = +) = \frac{3607}{9192}$
$P(Y_8 = +) = \frac{5067}{16384}$
$P(Y_9 = +) = \frac{5067}{32768}$

Question 2.1

See previous page.

Question 2.2

See page 4.

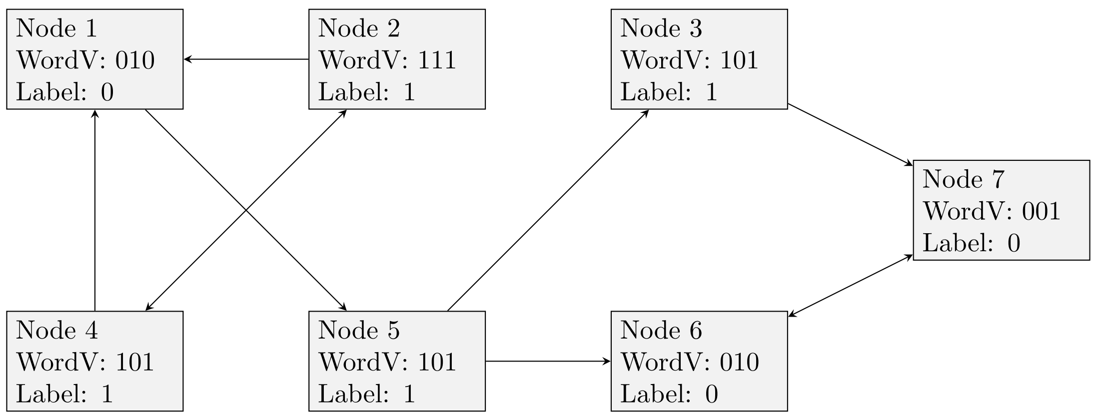
Question 2.3

See page 4.

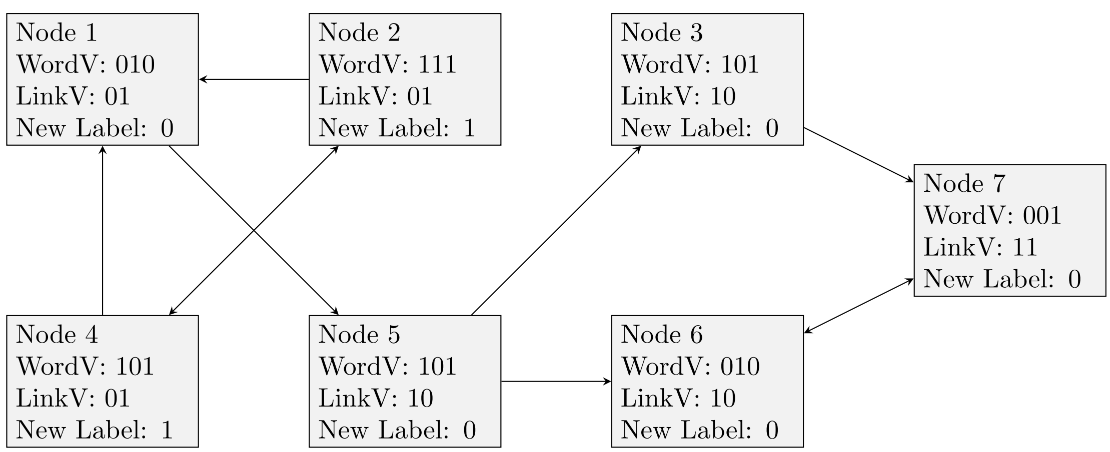
Question 2.4

$1, 2, 4, 5, 8, 9$

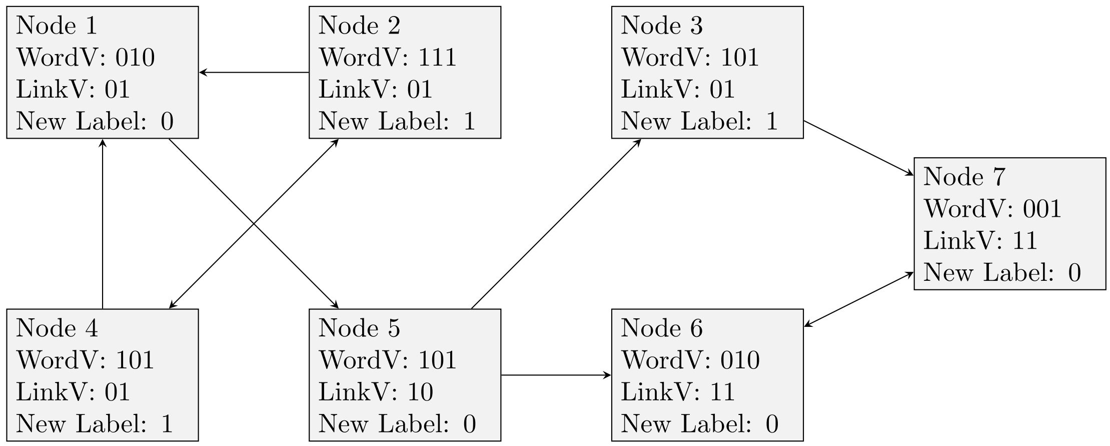
Question 3.1



Question 3.2



Question 3.3



Question 3.4

The label of node 3 changed from 0 to 1. This implies that LinkV will change to 10 in the next iteration. Due to the nature of g, this will not change the label of node 7. Since node 7 is the only node affected by node 3 the algorithm converges at the next step.

Question 4.1

We expect the answer to be 3 since the 3–hop neighborhood of the node is different but the 2–hop neighborhood is not. We can verify this with the following code.

Listing 1: Code to verify 4.1.

```
import numpy as np
import networkx as nx

G = nx.Graph()
H = nx.Graph()

for i in range(1, 8):
        G.add_node(i, label=i)
        H.add_node(i, label=i)

G.add_edges_from([
        (1, 2),
        (2, 3),
        (3, 4),
        (4, 5),
        (5, 6),
        (6, 7),
        (7, 2),
])
H.add_edges_from([
```

```
            (1, 2),
            (2, 3),
            (3, 4),
            (4, 5),
            (5, 6),
            (7, 2),
])

A = nx.adjacency_matrix(G)
B = nx.adjacency_matrix(H)
I = np.diag(np.ones(7))
h = np.ones(7)

h1 = h*(A+I)
h2 = h*(B+I)
i = 1
while h1.A[0][0] == h2.A[0][0]:
            h1 = h1*(A+I)
            h2 = h2*(B+I)
            i += 1
```

Question 4.2

i. The matrix M is given by $M = D^{-1}A$ where $D = \text{diag}(1/2, 1/2, 1, 1/3)$ is the diagonal matrix of (in)degrees and

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Thus,

$$M = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \end{pmatrix}.$$

ii. This can, for example, be computed by

Listing 2: Code to compute r.

```
import numpy as np

A = np.array([[0, .5, 0, 1/3], [.5, 0, 0, 1/3], [0, 0, 0, 1/3], [.5, .5, 1, 0]])
np.linalg.eig(A)
```

so that r is given by the vector $[0.47, 0.47, 0.24, 0.71]$.

Question 4.3

Assume that the graph has n nodes and that $h_i^{(l)}$ is a column vector. Put

$$h^{(l)} = \begin{pmatrix} | & & | \\ h_1^{(l)} & \dots & h_n^{(l)} \\ | & & | \end{pmatrix}.$$

We have

$$\begin{aligned} h_i^{(l+1)} &= \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} h_j^{(l)} \\ &= \frac{1}{|\mathcal{N}_i|} h^{(l)} \begin{pmatrix} a_{i1} \\ \vdots \\ a_{in} \end{pmatrix}. \end{aligned}$$

Now $\frac{1}{|\mathcal{N}_i|} \begin{pmatrix} a_{i1} & \dots & a_{in} \end{pmatrix}^\top$ is the i–th column of $A^\top D^{-1}$, so that $h^{(l+1)} = h^{(l)} A^\top D^{-1}$. So the transition matrix M of the random walk can be written as $M = D^{-1}A$.

Question 4.4

With the same notation as in 4.3 we have

$$\begin{aligned} h_i^{(l+1)} &= \frac{1}{2} h_i^{(l)} + \frac{1}{2|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} h_j^{(l)} \\ &= \frac{1}{2} h_i^{(l)} + \frac{1}{2|\mathcal{N}_i|} h^{(l)} \begin{pmatrix} a_{i1} \\ \vdots \\ a_{in} \end{pmatrix}. \end{aligned}$$

Again, the second term is given by the i–th column of $\frac{1}{2} h^{(l)} A^\top D^{-1}$ and the first term is the i–th column of $\frac{1}{2} h^{(l)} I$ with I being the $n \times n$ identity matrix. This implies

$$h^{(l+1)} = \frac{1}{2} h^{(l)} (A^\top D^{-1} + I)$$

and hence the transition matrix is given by $\frac{1}{2}(D^{-1}A + I)$.

Question 4.5

Let $e = (1, 1, \dots, 1)$ be a stationary distribution of our Markov chain described by the matrix $M = A^\top D^{-1}$, i.e. $eM = e$. Since the graph is connected the chain is irreducible and since the graph does not have any bipartite components the chain is aperiodic. By the Markov convergence theorem, all the rows of

M^l converge to e as $l \to \infty$. But e has all entries equal to 1 and $h^{(l+1)} = h^{(l)} A^\top D^{-1} = h^{(l)} M = h^{(0)} M^l$, so that for large values of l

$$h^{(l)} \approx h^{(0)} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

and

$$h_i^{(l)} \approx \sum_{j=0}^{n} h_j^{(0)}.$$

In other words, the node embeddings of all nodes will look very similar for large l.

Question 4.6

Let s be the source node. For each node u we set

$$h_u^{(l+1)} = \max\{h_v^{(l)}; v \in (u) \cup \{v\}\}$$

and $h_u^{(0)}$ for all nodes u but s and $h_s^{(0)} = 1$. We show that the GNN then learns BFS perfectly by induction. Assume that the hypothesis is correct at step l. If node u is visited at step $l+1$ of a BFS, then some neighbor v of u was visited at step l and thus $h_v^{(l)} = 1$. Therefore, $h_u^{(l+1)} = 1$. If node u has not been visited and is *not* visited at step $l+1$ of a BFS, then $h_v^{(l)} = 0$ for all $v \in \mathcal{N}(u)$ and $h_u^{(l+1)} = 0$. If node u has previously been visited, then clearly $h_u^{(l+1)} = 1$. So the induction hypothesis is true at step $l+1$ and the GNN learns BFS perfectly.

Question 5.1

The decoder is the inner product.

Question 5.2

If we put

$$Z = \begin{pmatrix} | & & | \\ z_1 & \dots & z_n \\ | & & | \end{pmatrix},$$

then the objective becomes

$$\min_Z \left\| A - Z^\top W Z \right\|_2 .$$

Question 5.3

The matrix W would need to be a diagonal matrix with the eigenvalues of A along the diagonal.

Question 5.4

Since A^i holds information about which nodes are connected through i hops the corresponding matrix factorization problem would be

$$\min_Z \left\| \sum_{i=1}^{k} A^i - Z^\top Z \right\|_2 .$$

Question 5.5

The embeddings of the 10–node cliques will be very similar since the random walk is very likely to move within a clique.

The vector representations of the nodes will encourage structural similarity within the cliques because a random walk is unlikely to go away from the 2–hop neighborhood of a starting node. The inout parameter will only affect the nodes at the bridge. So the nodes $N_R(u)$ visited by the biased walk are very similar within a clique. The random walks therefore will not catch the structural similarity of nodes that lie in two different cliques.

Question 5.6

The neighbors of w are all in the same clique so that only these can be reached.

Due to $g_k(u,v) > 0$ we can reach all nodes of the same blue color as w, i.e. the nodes of both cliques minus the ones of the bridge.

Question 5.7

If, for example, we would only use a value of 2 or 3, then the algorithm would not catch the higher order neighborhood structure of the nodes. This would imply that the gray and purple nodes would have embeddings that are very close together.

Question 5.8

Since struc2vec catches the structural similarity between the nodes, the vector representations of the two cliques should be very close together for the blue nodes and the one of the two green nodes should be close to this cluster but a bit apart from it.