Homework 1

Victor Croisfelt
victorcroisfelt@gmail.com

January 11, 2022

1 Link Analysis

Let $\mathbf{v}_i \in \mathbb{R}^d$ denote the PageRank vector for a user $i \in \mathcal{S}$, where d is the dimension of the graph of webpages, and $\mathcal{S}$ is the set of users. For each user, we perform the PageRank algorithm with a personalized teleport in order to obtain $\mathbf{v}_i$. For the i-th user, the following expression defines the PageRank with teleport:

$$\mathbf{v}_i = \underbrace{(\beta \mathbf{M} + (1-\beta)\mathbf{T}^i)}_{\text{"Google matrix"}} \mathbf{v}_i, \tag{1}$$

where $\beta \in [0, 1]$ defines the probability of non-teleport, $\mathbf{M} \in \mathbb{R}^{d \times d}$ is the left stochastic matrix, and $\mathbf{T}^i \in \mathbb{R}^{d \times d}$ is the teleport matrix, which defines how the i-th user personalizes the teleport process. For example, the given teleport set of user $A \in \mathcal{S}$ is $\{1, 2, 3\}$, meaning that only the entries of $\mathbf{T}^i$ related to indices $1, 2, 3$ are non-zero. From now on, we assume fixed β and that the teleport is uniform, meaning equal probabilities to access the webpages in the teleport set. For example, for user A, the probability to access each one of the webpages in $\{1, 2, 3\}$ is $\frac{1}{3}$.

First, we start by rewriting (1) as:

$$\mathbf{v}_i = \beta \mathbf{M} \mathbf{v}_i + (1-\beta)\mathbf{T}^i \mathbf{v}_i. \tag{2}$$

We define $\mathbf{p}_i = (1-\beta)\mathbf{T}^i \mathbf{v}_i$ as the contribution to the PageRank vector, $\mathbf{v}_i$, made by the teleport procedure. Then, we have

$$\begin{aligned}
\mathbf{v}_i &= \beta \mathbf{M} \mathbf{v}_i + \mathbf{p}_i && (3)\\
\mathbf{v}_i - \beta \mathbf{M} \mathbf{v}_i &= \mathbf{p}_i && (4)\\
\underbrace{(\mathbf{I}_d - \beta \mathbf{M})}_{\mathbf{B} \in \mathbb{R}^{d \times d}} \mathbf{v}_i &= \mathbf{p}_i && (5)\\
\mathbf{B} \mathbf{v}_i &= \mathbf{p}_i. && (6)
\end{aligned}$$

Note that the above expression is a linear system. Generally, $\mathbf{B}$ is unknown, since we do not have access to the graph of webpages, meaning that $\mathbf{M}$ is not available. However, $\mathbf{p}_i$ is known, since the PageRank vector $\mathbf{v}_i$ and the teleport set are known. We then make the following observation.

Remark 1.1. Without loss of generality, observe that if we consider no teleport procedure, the linear system becomes $\mathbf{B}\mathbf{v}_i = \mathbf{0}$, which is a *homogeneous system*. In this very particular case, the PageRank vectors $\mathbf{v}_i$ of different users in $\mathcal{S}$ are equal, because there is no personalization. We denote this PageRank vector as $\mathbf{v}$, which is the solution of the homogeneous system.

Remark 1.2. Now, with the teleport procedure, we have a nonhomogeneous system. For an arbitrary value of $\mathbf{p}_i$, the general solution of the nonhomogeneous system is thus in the solution set $\{\mathbf{v} + \mathbf{v}_i\}$, assuming that the nonhomogeneous system has at least one solution given by $\mathbf{v}_i$. The solution set can be written as $\{\mathbf{v} + (1-\beta)^{-1}(\mathbf{T}^i)^{-1}\mathbf{p}_i\}$ by considering that $\mathbf{T}^i$ has inverse. Therefore, given a subset $\{A, B, C, D\} \subset \mathcal{S}$ of users, their solution only differ due to the difference among their teleport sets (or teleport matrices), meaning that they have different $\mathbf{p}_i$'s.

Given that we know the PageRank vectors ($\mathbf{v}_i$'s) and the teleport sets (embedded in $\mathbf{p}_i$'s) of a subset of users $\mathcal{S}' \subset \mathcal{S}$, we can then find the PageRank vector of a user $k \in \mathcal{S} \setminus \mathcal{S}'$ if and only if its teleport set ($\mathbf{p}_k$) can be written as combination of the teleport sets of other users ($\mathbf{p}_i$ for $i \in \mathcal{S}'$). The reason why this is true is that the linear system regarding the k-th user can be seen as a translation of the linear systems of the other users with different $\mathbf{p}_i$'s. Moreover, part of the solution set for all users is similar due to the homogeneous contribution.

To write the above conclusion in a more formal way, we introduce the vector notation to represent the webpages in the teleport set of a user in the following. Given a known teleport set, the teleport vector that represents this set is given by $\mathbf{u}_i \in \{0,1\}^N$ for $i \in \mathcal{S}$, where the j-th element of $\mathbf{u}_i$ is 1 if the j-th webpage is contained in the teleport set, and 0 otherwise. The dimension of the teleport vector is given by N, which is the maximum webpage index considering the union of all teleport sets. Therefore, the PageRank vector $\mathbf{v}_k$ of a new user $k \in \mathcal{S} \setminus \mathcal{S}'$ can be obtained from a known subset of users $\mathcal{S}'$ if and only if

$$\mathbf{u}_k \in \text{span}(\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{|\mathcal{S}'|}\}), \tag{7}$$

where span denotes the space spanned by the vector space $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{|\mathcal{S}'|}\}$ (linear combination). This means that the solution of the linear system $\mathbf{B}\mathbf{v}_k = \mathbf{p}_k$ of the new, unknown user can be obtained by using knowledge about the solution of the linear systems of other known users without explicitly knowing the coefficient matrix $\mathbf{B}$ (the graph of webpages).

We can now use the introduced notation and conclusion above to evaluate the proposed questions. In our case, we known the PageRank vectors and teleport sets of 4 users, which are given in set notation as $\mathcal{S}' = \{A, B, C, D\} \subset \mathcal{S}$. The teleport sets are respectively: $\{1, 2, 3\}$, $\{3, 4, 5\}$, $\{1, 4, 5\}$, and $\{1\}$; hence, the largest index of a webpage is $N = 5$. These teleport sets can be represented in matrix form as:

$$\mathbf{U} \in \{0,1\}^{(N=5)\times(|\mathcal{S}'|=4)} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \tag{8}$$

where $\mathbf{U} = [\mathbf{u}_A, \mathbf{u}_B, \mathbf{u}_C, \mathbf{u}_D]$. Thereby, we can answer the first part of the problem:

"Assume that the weights for each node in a teleport set are uniform. Without looking at the graph (i.e. actually running the PageRank algorithm), can you compute the personalized

PageRank vectors for the following users? If so, how? If not, why not? Assume a fixed teleport parameter β."

Yes, it is possible. How: we need to evaluate if the personalized set of the unknown user is a linear combination of the teleport sets of known users.

Question 1.1 – Personalized PageRank I

Eloise's teleport vector is $\mathbf{u}_E = [0, 1, 0, 0, 0]^\intercal$. The augmented matrix $[\mathbf{U}|\mathbf{u}_E]$ is

$$[\mathbf{U}|\mathbf{u}_E] = \left[\begin{array}{cccc|c} 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array}\right]. \tag{9}$$

The rank$(\mathbf{U}) = 4$, while rank$([\mathbf{U}|\mathbf{u}_E]) = 4$. This means that the system is consistent and $\mathbf{u}_E$ can be written as a linear combination of the teleport vectors of the known users. Therefore, we **can** compute the personalized PageRank vector $\mathbf{v}_E$ of Eloise in an indirect way. By analysing the augmented matrix above, the relationship is:

$$\mathbf{v}_E = \mathbf{v}_A - \mathbf{v}_B - \mathbf{v}_C. \tag{10}$$

Question 1.2

First, let us represent the new teleport matrix $\tilde{\mathbf{U}} \in \{0, 1\}^{(N=5)\times(|\mathcal{S}'|=5)}$ from the known users, now including Eloise, as:

$$\tilde{\mathbf{U}} = \left[\begin{array}{ccccc} 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array}\right]. \tag{11}$$

Felicity's teleport vector is $\mathbf{u}_F = [0, 0, 0, 0, 1]^\intercal$. The augmented matrix $[\tilde{\mathbf{U}}|\mathbf{u}_F]$ is

$$[\tilde{\mathbf{U}}|\mathbf{u}_F] = \left[\begin{array}{ccccc|c} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]. \tag{12}$$

The rank$(\tilde{\mathbf{U}}) = 4$, while rank$([\tilde{\mathbf{U}}|\mathbf{u}_F]) = 5$. This means that the system is inconsistent and the PageRank vector of Felicity **cannot** be obtained without the knowledge of the graph of webpages.

Question 1.3

To enable the use of *weights*, we drop the assumption that the representations of teleport sets are defined as $\mathbf{u}_i \in \{0,1\}^N$. Thus, we assume the more general definition of $\mathbf{u}_i \in \mathbb{R}^N$. Since Glynnis's teleport vector is $\mathbf{u}_G = [0.1, 0.2, 0.3, 0.2, 0.2]^\intercal$, the augmented matrix $[\tilde{\mathbf{U}}|\mathbf{u}_G]$ is

$$[\tilde{\mathbf{U}}|\mathbf{u}_G] = \left[\begin{array}{ccccc|c} 1 & 0 & 1 & 1 & 0 & 0.1 \\ 1 & 0 & 0 & 0 & 1 & 0.2 \\ 1 & 1 & 0 & 0 & 0 & 0.3 \\ 0 & 1 & 1 & 0 & 0 & 0.2 \\ 0 & 1 & 1 & 0 & 0 & 0.2 \end{array}\right]. \tag{13}$$

The $\text{rank}(\tilde{\mathbf{U}}) = 4$, while $\text{rank}([\tilde{\mathbf{U}}|\mathbf{u}_G]) = 4$. This means that the system is consistent and the PageRank vector of Glynnis **can** be obtained without the knowledge of the graph of webpages. Note that $\tilde{\mathbf{U}}$ is singular and we have more equations than unknowns, the last two rows are similar. Thus, using least squares, the solution is

$$\mathbf{v}_G = 0.125\mathbf{v}_A + 0.175\mathbf{v}_B + 0.025\mathbf{v}_C - 0.05\mathbf{v}_D + 0.075\mathbf{v}_E. \tag{14}$$

Question 1.4 – Personalized PageRank II

The set of all personalized PageRank vectors that you can compute from $\mathcal{V}$ without accessing the web graph is given by:

$$\text{span}(\mathcal{V}) = \{\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_{|\mathcal{V}|}\mathbf{v}_{|\mathcal{V}|}\}, \tag{15}$$

which represents the set of all vectors that can be written as a linear combination of the vectors in $\mathcal{V}$, $\forall\alpha_i \in \mathbb{R}$.

Question 1.5 – A different equation for PageRank

From $\mathbf{r} = \mathbf{A}\mathbf{r}$ and the definition of $\mathbf{A} = \beta\mathbf{M} + \frac{(1-\beta)}{N}\mathbf{1}\mathbf{1}^\intercal$, we have

$$\begin{aligned} \mathbf{r} &= \left(\beta\mathbf{M} + \frac{(1-\beta)}{N}\mathbf{1}\mathbf{1}^\intercal\right)\mathbf{r} \\ &= \beta\mathbf{M}\mathbf{r} + \frac{(1-\beta)}{N}\mathbf{1}(\mathbf{1}^\intercal\mathbf{r}) \\ &\overset{(a)}{=} \beta\mathbf{M}\mathbf{r} + \frac{(1-\beta)}{N}\mathbf{1}\left(\sum_{i=1}^{N} r_i\right) \\ &= \beta\mathbf{M}\mathbf{r} + \frac{(1-\beta)}{N}\mathbf{1}, \qquad \blacksquare \end{aligned} \tag{16}$$

where in (a) we used the fact that $\mathbf{r}$ is a stationary distribution and, consequently, $\sum_{i=1}^{N} r_i = 1$.

2 Relational Classification I

Table 1: Evolution of the relational classifier

	iteration 1		iteration 2		label
node	$P^{(1)}(Y_i = -)$	$P^{(1)}(Y_i = +)$	$P^{(2)}(Y_i = -)$	$P^{(2)}(Y_i = +)$	y_i
1	0.5	0.5	0.5	0.5	0
2	0.625	0.375	0.6016	0.3984	0
3	0.375	0.625	0.3672	0.6328	1
4	0.5312	0.4688	0.5293	0.4707	0
5	1	0	1	0	0
6	0	1	0	1	1
7	0	1	0	1	1
8	0.5156	0.4844	0.6436	0.3564	0
9	0.7578	0.2422	0.8218	0.1782	0
10	0	1	0	1	1

Question 2.1

The result is $P(Y_3 = +) = 0.6328$.

Question 2.2

The result is $P(Y_4 = +) = 0.4707$.

Question 2.3

The result is $P(Y_8 = +) = 0.3564$.

Question 2.4

Following the results reported in Table 1 and after the realization of two iterations, the nodes that belong to class "-" are: {1,2,4,5,8,9}.

3 Relational Classification II

Question 3.1 – Bootstrap Phase

The label vector $\mathbf{y}^{(0)} \in \mathbb{R}^7$ is $[0, 1, 1, 1, 1, 0, 0]^\intercal$ after the realization of the Bootstrap phase, which is indexed by the superscript (0).

Question 3.2 – Iteration 1

Table 2 gives the most important results involving the **first iteration** of the method of iterative node classification. Changes in the label vector are marked with **blue** color.

Table 2: Iteration 1

node	pre-label $\mathbf{y}^{(0)}$	link vector $[I_0, I_1]^\intercal$	pos-label $\mathbf{y}^{(1)}$
1	0	$[0, 1]^\intercal$	0
2	1	$[0, 1]^\intercal$	1
3	1	$[0, 1]^\intercal$	1
4	1	$[0, 1]^\intercal$	1
5	1	$[1, 0]^\intercal$	**0**
6	0	$[1, 1]^\intercal$	0
7	0	$[1, 1]^\intercal$	0

Question 3.3 – Iteration 2

Table 3 gives the results for the **second iteration**.

Table 3: Iteration 2

node	pre-label $\mathbf{y}^{(1)}$	link vector $[I_0, I_1]^\intercal$	pos-label $\mathbf{y}^{(2)}$
1	0	$[0, 1]^\intercal$	0
2	1	$[0, 1]^\intercal$	1
3	1	$[1, 0]^\intercal$	**0**
4	1	$[0, 1]^\intercal$	1
5	0	$[1, 0]^\intercal$	0
6	0	$[1, 0]^\intercal$	0
7	0	$[1, 1]^\intercal$	0

Question 3.4 – Convergence

Yes. In Iteration 2, only the label of node 3 changed from 1 to 0. Node 7 is the only that suffers from this change, where its new link vector becomes $[1, 0]^\intercal$. However, the new link vector does not change the label of node 7 in a future realization of the algorithm, since $I_0 = 1 \implies y_7^{(3)} = 0 \equiv y_7^{(2)}$. Hence, the algorithm converged.

4 GNN Expressiveness

Question 4.1 – Effect of Depth on Expressiveness

For the given graphs and positions of red nodes, we need at least $K = 2$ hops so as we can start distinguish the two red nodes. This is because their neighborhood patterns are the same when considering a single hop. Such effect shows the importance of choosing the number of hops in order to better capture the difference in the structure of graphs. Not only the number of hops, but also the introduction of non-linearity and learnable parameters.

Question 4.2 – Random Walk Matrix

i) First, we obtain the following adjacency matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \tag{17}$$

From the adjacency matrix, we normalized its columns to sum up to 1 to obtain the *left stochastic matrix* $\mathbf{M}$:

$$\mathbf{M} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \end{bmatrix}. \tag{18}$$

ii) Instead of computing the eigenvector of $\mathbf{M}$ through its characteristic polynomial, we opted to take advantage of the fact that the stochastic distribution converges when performing $\mathbf{r} = \mathbf{M}(\mathbf{M}(\ldots(\mathbf{Mr})))$, where the system reaches its steady state. By doing so, we obtain the following stationary distribution:

$$\mathbf{r} = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{3}{8}\right], \tag{19}$$

The l_2-normalized stationary vector is $[\sqrt{2}/3, \sqrt{2}/3, 1/(3\sqrt{2}), 1/\sqrt{2}]^\intercal$. In fact, this result was obtained running the Power Iteration algorithm with tolerance error $\epsilon = 10^{-6}$.
NOTE: Only in this Question 4.2, we are abusing of the notation since $\mathbf{r}$ is a row vector! Read more about the use of left and right stochastic matrices in https://en.wikipedia.org/wiki/Stochastic_matrix. Below, we use the adequate right stochastic version of it. A similar derivation could define as a doubly stochastic matrix, which would originate expressions like the one presented in the question: $D^{-1/2}AD^{-1/2}$

Question 4.3 – Relation to Random Walk (i)

Since we are considering a *1-hop-uniform* Random Walk, with the adjacency matrix $\mathbf{A}$ in (17), we have that

$$\mathbf{h}^{(l+1)} = \mathbf{A}\mathbf{D}^{-1}\mathbf{h}^{(l)}, \tag{20}$$

where $\mathbf{D} = \text{diag}([D_{11}, D_{22}, \ldots, D_{|\mathcal{V}||\mathcal{V}|}])$ is the diagonal matrix with the degree of each node and $\mathbf{h}^{(l)} \in \mathbb{R}^{|\mathcal{V}|}$ is the *column* vector of Random Walk probabilities at layer l. Thus, the *transition matrix* is $\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$, the elements of this matrix is called as transition probabilities. The u-th element of $\mathbf{h}$

Question 4.4 – Relation to Random Walk (ii)

In matrix form, from (20), the aggregation operation can be described as:

$$\mathbf{h}^{(l+1)} = \frac{1}{2}\mathbf{h}^{(l)} + \frac{1}{2}\mathbf{A}\mathbf{D}^{-1}\mathbf{h}^{(l)} \tag{21}$$

$$= \frac{1}{2}\left(\mathbf{I}_{|\mathcal{V}|} + \mathbf{A}\mathbf{D}^{-1}\right)\mathbf{h}^{(l)} \tag{22}$$

where the transition matrix is now $\tilde{\mathbf{P}} = \frac{1}{2}\left(\mathbf{I}_{|\mathcal{V}|} + \mathbf{A}\mathbf{D}^{-1}\right)$.

Question 4.5 – Over-Smoothing Effect

To answer this question, we are going to outline the steps related to the Convergence Theorem for Finite Markov Chains. In doing so, we first need to define some concepts based on [1].

Definition 4.1. A **finite Markov chain** with **finite state space** $\mathcal{V}$ (the set of nodes of a graph G) and left transition matrix $\mathbf{P}$ is a sequence of random variables $X_0, X_1, \ldots$ where

$$\mathbb{P}\{X_{t+1} = x | X_t = y\} = P_{x,y},$$

where t represents a given discrete time instant and $x, y \in \mathcal{V}$. The probability of an event is denoted as $\mathbb{P}$.

Definition 4.2. A **probability distribution** or just a **distribution** is a *column* vector of non-negative values that sums up to 1.

Definition 4.3. A distribution $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{V}|}$ is a stationary distribution of a Markov chain defined by the transition matrix $\mathbf{P}$ if $\boldsymbol{\pi} = \mathbf{P}\boldsymbol{\pi}$.

The above definition means that, at a given time instant t, the distribution at state X_t is the same as the one at state X_{t+1}. Therefore, no matter where you are, the distribution of moving along the Markov chain does not change the probability of being at a particular state. This is called as the steady-state condition of the Markov chain. However, the definition of such distribution is related to the following questions:

1. Is the stationary distribution **unique**?
2. **When** a Markov chain will **have** a stationary distribution?
3. Given an initial distribution, will it **converge** to the stationary distribution over time?

The questions above are answered in proving the following facts about the stationary distribution $\boldsymbol{\pi}$ in the given order [1]:

1. **Uniqueness:** A stationary distribution $\boldsymbol{\pi}$ is unique if $\mathbf{P}$ is *irreducible*.

2. **Existence:** Based on the irreducible principle and the definition of hitting time (first return time), it is possible to explicitly construct a stationary distribution. Due to the uniqueness property stated above, this stationary distribution is unique for the given irreducible $\mathbf{P}$. Basically, the obtained stationary distribution defines its probability by the inverse of the expected hitting time of a particular state. Formally,

$$\pi_x = \frac{1}{\mathbb{E}\{\tau_x^+\}},$$

   where $\tau_x^x = \min\{t > 0 : X_t = x\}$ is the hitting time of state x.

3. **Convergence:** Given that $\mathbf{P}$ is irreducible, it is possible to show that any distribution of the Markov chain will convergence to the stationary distribution defined/obtained above if and only if the the Markov chain is also *aperiodic*. The convergence is based on the total variation, which is a distance metric that evaluates the distance between distributions (stationary and any).

Given the outline of the main results above, we need to define what is the meaning of a Markov chain being irreducible and aperiodic.

Definition 4.4. A Markov chain is **irreducible** if for all states $x, y \in \mathcal{V}$, there exists a $t \geq 0$ such that $P_{x,y}^t > 0$. This means that, given a finite amount of steps, it is always possible to get from x to y, for any states $x, y \in \mathcal{V}$.

Definition 4.5. Let $\mathcal{T}(x) = \{t \geq 1 : P_{x,x}^t > 0\}$ be the set of all time steps for which a Markov chain can start and end in a state x. Then the period of x is the $\gcd \mathcal{T}(x)$ (gcd–greatest common divisor).

From the above definition, it is possible to show that the period of all states of an irreducible Markov chain is the same. Thus, an aperiodic and irreducible Markov chain is the one that has period equal to 1, or, $\gcd \mathcal{T}_x = 1$, $\forall x \in \mathcal{V}$. Intuitively, this means that there is no preferred state over time; at a given time, all the states can be reached.

With this small recap of the Convergence Theorem for Finite Markov Chains, we see that if $\mathbf{P}$ is irreducible and aperiodic, any distribution converges exponentially to the stationary distribution. The exponential notion comes from the fact that the convergence depends on $\mathbf{P}^t$, which represents the realization of t steps along the Markov chain. Therefore, in the limit of $t \to \infty$, we will have that the embedding of two different nodes tend to converge to the stationary distribution $\boldsymbol{\pi}$.

Question 4.6 – Learning BFS with GNN

The most important part of the breadth-first search (BFS) algorithm is its queue, whose order is defined by the following phrase:

"nodes that are connected to already visited nodes become visited".

Therefore, a possible way to learn the BFS algorithm via GNN is with the following structure:

- **Initialization:** Only a source node $h_s^0 \leftarrow 1$ for $s \in \mathcal{V}$; the others receive $h_n^0 \leftarrow 0$ for $n \in \mathcal{V} \setminus \{s\}$.
- **Message function:** Given a node v, its k-hop neighbors will send their embeddings $h_u^{(k)}$'s containing if they were visited (1) or not (0), for $u \in \mathcal{N}_v$. The message coming from node $u \in \mathcal{N}_v$ is then: $m_u^{(k)} = h_u^{(k-1)}$.
- **Aggregation:** For $v \in \mathcal{V}$, we first define the vector:

$$\mathbf{h}^{(k)} = \text{CONCAT}(h_v^{(k-1)}, \{m_u^{(k)}, \forall u \in \mathcal{N}_v\}).$$

  Then, the final aggregation is performed as:

$$h_v^{(k)} = \frac{\exp^{\text{avg}(\mathbf{h}^{(k)})}}{1 + \exp^{\text{avg}(\mathbf{h}^{(k)})}}$$

  with the sigmoid function and average is performed over the node degree and avg denoting the computation of the average.
- **Update:** Finally, the update is

$$h_v^{(k)} \leftarrow \begin{cases} 1, & \text{if } h_v^{(k)} \geq 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

The above procedure is inspired on logistic regression; interpreting the visiting condition of a node as a classification problem.

5 Node Embedding and its relation to matrix factorization

Question 5.1 – Simple matrix factorization

The decoder is the inner product $\mathbf{Z}^\intercal\mathbf{Z} \approx \mathbf{A}$, which approximately gives back the adjacency matrix that describes the whole graph information.

Question 5.2 – Alternate matrix factorization

The objective would be $\min\|\mathbf{A} - \mathbf{Z}^\intercal\mathbf{W}\mathbf{Z}\|_2$, where $\mathbf{W}$ acts as a weight matrix.

Question 5.3 – Relation to eigendecomposition

From the definition of eigenvalue and because the adjacency matrix is square, we have that:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \tag{23}$$

where λ is the eigenvalue associated with the eigenvector $\mathbf{v}$. By assuming that the adjacency matrix has a set of $|\mathcal{V}|$ linearly independent eigenvectors and by using that the eigenvectors are orthogonal to each other, we have that:

$$\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}, \tag{24}$$

where $\mathbf{Q} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{|\mathcal{V}|}]^\intercal$ and $\mathbf{\Lambda} = \text{diag}([\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{V}|}])$. By left-multiplying $\mathbf{Q}^{-1}$ on the expression above, we obtain the following factorization called as eigendecomposition:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}. \tag{25}$$

By matching the above expression with the approximation $\mathbf{Z}^\intercal\mathbf{W}\mathbf{Z}$, the weight matrix $\mathbf{W}$ should meet the eigenvalue matrix $\mathbf{\Lambda}$ of the adjacency matrix $\mathbf{A}$, which makes sense; (personal note: there is a similar result for estimation theory, for example).

Question 5.4 – Multi-hop node similarity

Assume the multi-hop definition:

"2 nodes are similar if they are connected by at least one path of length at most k",

where k is a parameter. The corresponding matrix factorization problem would be then:

$$\|\mathbf{A}^k - \mathbf{Z}^\intercal\mathbf{W}\mathbf{Z}\|, \tag{26}$$

where $\mathbf{A}^k$ describes the k-th order connections of nodes.

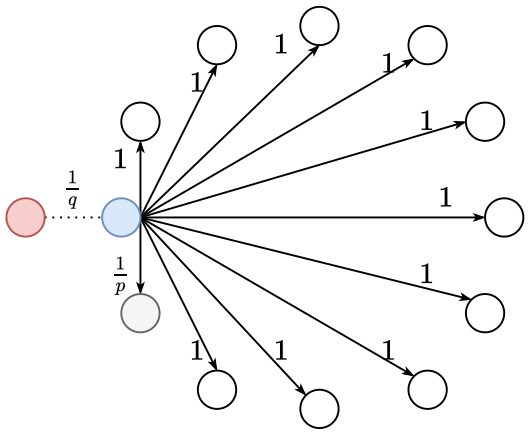

Figure 1: Characterization of Node2vec. The neighborhood of the blue node is being evaluated. The gray node denotes the preceding node, while the red node corresponds to the red node given in the homework's figure. The dot line represents the condition of the green node in the homework's figure, which connects the 10-node clique to the bridge.

Question 5.5 – Limitation of node2vec (i)

Fig. 1 characterizes the biased fixed-length Random Walk defined by the node2vec algorithm for the given 10-nodes clique. We suppose that the neighborhood of the blue node is being evaluated. Since the node2vec considers the 2-hop condition, without loss of generality, we assume that the gray node precedes the choice of the blue one. In the very special case of the green node in the homework's figure, we have to consider the dotted link.

Based on the fact that all the nodes within a clique are connected with each other, the numbers associated to the links in Fig. 1 denote the unnormalized biased probabilities to conduct the Random Walk given the current position at the blue node. The meanings are:

- 1 is assigned to nodes that are 1-hop distant to the previous node. Recall that withing a clique all nodes are connected;
- $1/p$ is the probability of going back to the previous (gray) node, p is called the return parameter;
- $1/q$ is the probability of going to a node that is more than 1-hop farther from the previous node, q is called the in-out parameter.

With the above characterization, we are able to answer the question:

"Do you think the node embeddings will reflect the structural similarity? Justify your answer."

No, I do not. To begin with, it is very likely that the node embeddings of all the nodes within a clique will be almost the same; irrespective of their positions. This is because all the nodes are connected with each other, resulting in a high probability of staying in the local neighborhood – within the 10-nodes clique. Now, the notion of **structural similarity** represents the case in which one wants to embed two nodes that have the same types of

connections and play a role similar in the graph together, even though they are positioned in different parts of the graphs. However, with node2vec, the nodes in the two different cliques will end up with different embeddings, meaning that structural similarity is not capture at all. This is because it is very unlikely to cross the bridge and capture similarity between the two 10-nodes clique structure. For example, if we want to extract structural similarity, green nodes (homework's figure – blue one for us) must have a similar node embedding, since they have a similar structural function. However, since similarity in node2vec is based on co-occurrence of nodes given the realization of a Random Walk, it is very difficult to have the two green nodes co-occurring in a Random Walk to try to characterize this structural similarity. In the bridge, the probabilities will be dependent on $1/p$ and $1/q$ and they will be fighting against each other.

To summarize: node2vec similarity notion is based on the co-occurrence of nodes during a Random Walk. This similarity notion does not take into account structure/role of the nodes. It just embeds nodes closer to each other together, where closeness depends on the choice of parameters q and p.

Question 5.6 – Limitation of node2vec (ii)

Fig. 1 answers the question for the node2vec. The node2vec algorithm is based on a 2nd-order similarity measure, meaning that the Random Walk stores the preceding position of the walk and chooses the probabilities of the next walk based on the 2nd order distance taking the preceding node as reference. For the struct2vec algorithm, the probabilities are not limited by the notion of preceding node and 2nd order distances. In fact, the metrics $g_k(u, v)$ depends on the parameter K, which defines a distance to be considered; the larger K, the farther the walk can jump. Moreover, from a node u, you can go to any other node at a distance K given the weight probabilities $g_k(u, v)$. This means that you can avoid sticking within a local neighborhood with higher probability.

Question 5.7 – Limitation of node2vec (iii)

Part of it was answered in the question above. The importance of having different g_k's is that the Random Walk is able to exploit nodes much farther away than the actual node, avoiding staying in a local neighborhood and allowing the detection of similarity among nodes that are located at very different places.

Question 5.8 – Limitation of node2vec (iv)

Fig. 2 tries to characterize the struc2vec approach for the green node. The idea is that the parameter K defines different tiers in which the weights $g_k(u, v)$ will change, meaning that you can reach a broader quantity of nodes compared to the node2vec. For example, for the graph shown in the question, when $K \geq 11$, it is possible to start to reach the second 10-nodes clique from the first one through the realization of a Random Walk in the first one. When computing the node embedding, the choice of between node2vec and struct2vec changes the set of neighbors $\mathcal{N}_R(u)$. Because of this, struct2vec is able to evaluate a little the second

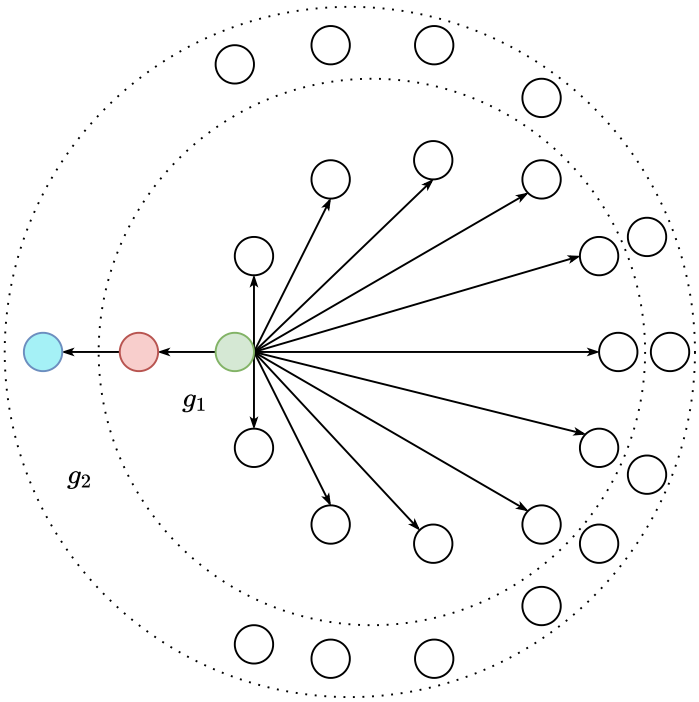

Figure 2: Characterization of struct2vec. Connections of the second tiers are omitted ($K = 2$).

clique in order to define the embedding of the ones within the first or vice versa. It can better capture the notion of structural similarity.

References

[1] Ari Freedman (2017) *Convergence Theorem for Finite Markov Chains.*