

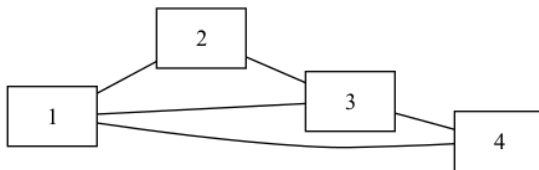
# Assignment 2

Barak-Nadav Diker

September 5, 2023

## Question 1

Given the following graph



Calculate all walks of size 6 in the given graph

## Answer

In order to do so we'll use the following theorem

**Theorem 1** ( Walks Theorem ). *If  $A$  is the adjacency matrix of a graph or digraph  $G$  with vertices  $\{v_1, \dots, v_n\}$ , then the  $i, j$  entry of  $A^k$  is the number of walks of length  $k$  from  $v_i$  to  $v_j$*

By Walks Theorem we can just multiple the adjacency matrix by itself 6 times and we'll get all the walks available from node  $i$  into node  $j$

Since the problem specify undirected graph we'll have to sum all elements of the matrix and divide it by 2

---

### Algorithm 1 All walks of length 6

---

```
 $n \leftarrow 6$  ▷ 6  $\Rightarrow$  length of walk  
 $Adj$  ▷ adjacency matrix  
 $M \leftarrow I$   
while  $n \neq 0$  do  
     $M \leftarrow M \times Adj$  ▷ Matrix Multiples  
end while  
 $sum \leftarrow 0$   
while  $a \in M$  do  
     $sum \leftarrow sum + a$   
end while
```

---

We will apply **algorithm 1** for getting the number of walk of length 6

```
import numpy  
from numpy.linalg import matrix_power  
input_array = numpy.array([[0,1,1,1],  
                             [1,0,1,0],  
                             [1,1,0,1],  
                             [1,0,1,0]])  
A_to_power6 = matrix_power(input_array, 6)  
sum_var = 0  
for row in A_to_power6:  
    for elem in row:  
        sum_var += elem  
int(sum_var/2)  
# output is 557
```

## Question 2

Consider the following quote

“Undirected Graph can be considered as directed graph”

Prove it

Formally , Given an Undirected graph find a directed graph such that  $A_G = A_{G'}$

## Answer

Given An undirected graph mark it  $G = (V, E)$  and the adjacency matrix of his as  $A_G$

Define the directed graph  $G' = (V', E')$  where  $V' = V$  and

$$\forall e = (v_1, v_2) \in E : e_1 = (v_1, v_2), e_2 = (v_2, v_1) \in E'$$

The adjancey matrix of the directed graph is equal to the adjancey matrix of the undirected graph

$$A_G = A_{G'}$$

## Question 3

Prove that given a directed graph  $G = (V, E)$  where  $V = (1, 2, \dots, n)$ , let A be the adjacency matrix

$$k \in \mathbb{N} \cup \{0\} : \forall i, j \in V : F(j, i, k) = (A^k)_{i,j}$$

where  $F : V \times V \times \mathbb{N} \cup \{0\} \rightarrow \mathbb{N} \cup \{0\}$  are all the walks from node j to i of length k

## Answer

We'll prove the theorem by induction

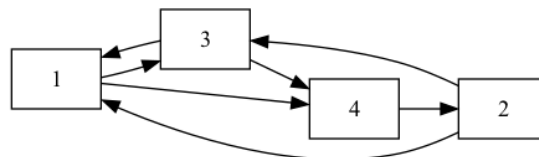
*Proof.* By induction

**Base Case:** For  $k = 1$ ,  $A^k = A$ , and there is a walk of length 1 between i and j if and only if  $a_{ij} = 1$ , thus the result holds.

**Step Case:** Assume the proposition holds for  $k = n$  and consider the matrix  $A_{n+1} = A_n A$ , By the inductive hypothesis, the  $(i, j)^{th}$  entry of  $A_n$  counts the number of walks of length n between vertices i and j. Now, the number of walks of length  $n + 1$  between i and j equals the number of walks of length n from vertex i to each vertex v that is adjacent to j. But this is the  $(i, j)^{th}$  entry of  $A^n A = A^{n+1}$  the non-zero entries of the column of A corresponding to v are precisely the first neighbours of v. Thus the result follows by induction on n  $\square$

## Question 4

Find the number of possible walks of length 8 from 1 to 4 by the following undirected graph



## Answer

We'll Write the adjacency matrix and calculate the  $A^8$   
To do so we'll use the same code

```
import numpy
from numpy.linalg import matrix_power
input_array = numpy.array([[0,1,1,0],
                           [0,0,0,1],
                           [1,1,0,0],
                           [1,0,1,0]])
A_to_power6 = matrix_power(input_array, 8)
A_to_power6[3][0]
# output is 23
```

To Calculate the Matrix by power of 8 we will use eigen values  $\det(A - \lambda I_n) = 0$  lets apply the calculation

$$\det(A - \lambda I_n) = \begin{vmatrix} -\lambda & 1 & 1 & 0 \\ 0 & -\lambda & 0 & 1 \\ 1 & 1 & -\lambda & 0 \\ 1 & 0 & 1 & -\lambda \end{vmatrix} = 0$$

In order to find eigen value will python code

```
import numpy as np
from numpy.linalg import eig
input_array = np.array([[0,1,1,0],
                        [0,0,0,1],
                        [1,1,0,0],
                        [1,0,1,0]])
w,v=eig(input_array)
w # The vector of eigen values
# w = | 1.69 | -1 | +0.j | -0.347-1.028j |
```

And finally we do

$$p^{-1}A^8p = \begin{pmatrix} \lambda_1^8 & 0 & 0 & 0 \\ 0 & \lambda_2^8 & 0 & 0 \\ 0 & 0 & \lambda_3^8 & 0 \\ 0 & 0 & 0 & \lambda_4^8 \end{pmatrix}$$

and change basis to get

$$A^8 = \begin{pmatrix} 19 & 23 & 18 & 13 \\ 13 & 14 & 13 & 10 \\ 18 & 23 & 19 & 13 \\ 23 & 26 & 23 & 14 \end{pmatrix}$$

## Question 5

Prove that the probability to pass from vertex  $j$  to vertex  $i$  is given by the matrix  $\tilde{A}_G = A_G * D_G^{-1}$  where the matrix  $D_G$  is define to be

$$D_G = \begin{pmatrix} \deg(1) & 0 & \dots & 0 \\ 0 & \deg(2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \deg(n) \end{pmatrix}$$

careful  $\tilde{A}_G$  might not be symmetric

## Answer

*Proof.* Since  $G$  is undirected graph and the probability to travel from  $j$  to  $i$  is even distributed, the probability matrix is to divide the matrix  $A_G$  column  $j$  by  $\deg(j)$  for each  $j \in \mathbb{N} \cap [1, n]$  or more formally

$$\forall j \in \mathbb{N} \cap [1, n], (\hat{A}_G)_{i,j} := \frac{(A_G)_{i,j}}{\deg(j)}$$

But Happily this is exactly equivalent to simply multiply the following matrix

$$\tilde{A}_G = A_G * D_G^{-1}$$

i.e

$$\tilde{A}_G = \hat{A}_G$$

Which is what we want it to be

□

## Question 6

let  $p^{(0)} \in \mathbb{R}$  be the initial probability distribution of the graph

let  $p^{(n)} \in \mathbb{R}$  be the distribution of the graph after  $n$  walks

prove that

$$p^{(n)} = \tilde{A}_G^n * p^{(0)}$$

## Answer

*Proof.* Given a vertex  $j$  and vertex  $i$  we first want to find all possible walks from  $j$  to  $i$  with length  $n$ ,

We have already proven that the number of possible walks are  $(A_G^n)_{i,j}$

In order to get the required probability we need to find the sum of all walks from  $j$  to any vertex i.e

$$\sigma_j = \sum_{i \in V} (A_G^n)_{i,j}$$

$$(Prob)_{i,j} = \frac{(A_G^n)_{i,j}}{\sigma_j}$$

Ultimately, I have proven that the probability of walking  $n$  walks from vertex  $j$  to  $i$  is simply  $\tilde{A}^n$

□

Note that we are not yet done we still have to prove that

$$p^{(n)} = \tilde{A}^n * p^{(0)}$$

To skip to that click final

Another proof is by induction

*Proof.*  $\tilde{A}^{m+1} = \tilde{A} * \tilde{A}^m$

**Base Case:** for  $n=0$  we have  $\tilde{A} = \tilde{A}$

**Step Case:** Given that the claim is true for  $n \in \mathbb{N}$  we will prove it for  $n+1$ , more specifically given  $j$  and  $i$  we are searching for the probability of going from  $j$  to  $i$  after  $n+1$  walks

Since by assumption we already know the probability of walking  $n$  long from  $j$  to any  $v \in V$  which is  $(A_G^n)_{v,j}$  and also we know the probability of getting from any vertex  $v$  into vertex  $i$  which is  $(A_G)_{i,v}$

The probability of getting from  $j$  to  $i$  after  $n+1$  walks is by conditional probability

$$Prob^{(n+1)} = \sum_{v \in V} (A_G)_{i,v} * (A_G^n)_{v,j}$$

since

$$P(B) = \sum_i P(B|A_i), \sum_i P(A_i) = 1$$

This equation is nothing but  $\tilde{A}^{m+1} = \tilde{A} * \tilde{A}^m$  □

Now Given a vertices  $i \in V$  by the complete probability theorem the probability of getting into vertex  $i$  is

$$p_i^{(n)} = \sum_{v \in V} \tilde{A}_{i,v} * p_v^{(0)}$$

But this equation is nothing but what we needed to prove which is

$$p^{(n)} = \tilde{A}^n * p^{(0)} \quad (1)$$

## Question 7

Prove that  $\tilde{A}$  is diagonalizable over  $\mathbb{R}$

### Answer

We will use the spectral theory for the prove  
Spectral Theory

**Reminder:** Please Note that  $\tilde{A} = A_G * D^{-1}$

By assumption , we know  $A$  is a symmetric matrix

lets mark the matrix

$$Q \in \mathbb{M}_n(\mathbb{R}) : Q := \text{diag}(\sqrt{\deg(v_1)}, \dots, \sqrt{\deg(v_n)})$$

Observe the following logic:

$$Q^{-1} * \tilde{A} * Q = Q^{-1} * A * D^{-1} * Q = Q^{-1} * A * Q^{-1}$$

where  $A = A^t$

$$\begin{aligned} (Q^{-1} * A * Q^{-1})^t &= (A * Q^{-1})^t * (Q^{-1})^t = \\ &= (Q^{-1})^t * A^t * (Q^{-1})^t \\ &= (Q^{-1})^t * A * (Q^{-1})^t \\ &= Q^{-1} * A * Q^{-1} \end{aligned}$$

By this equation we can infer that the matrix

$Q^{-1} * \tilde{A} * Q$  is a symmetric matrix which means that we can use on it the spectral theory

$$\begin{aligned} \exists P \in \mathbb{M}_n(\mathbb{R}) : P^{-1} * (Q^{-1} * \tilde{A} * Q) * P &= \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \\ &= (QP)^{-1} \tilde{A} QP \end{aligned}$$

So I have found a matrix  $QP$  which diagonalize the matrix  $\tilde{A}$

## Question 8

let  $\lambda$  be an eigen value of  $\tilde{A}$  proof that  $\lambda \in [-1, 1]$

### Answer

I'll prove it by contradiction , on the matrix  $\tilde{A}^t$ , let  $\lambda$  be eigen value of  $\tilde{A}$

$$\exists v \in V : \tilde{A}v = \lambda v$$

for some  $\lambda > 1$

Since the row of the matrix  $\tilde{A}^t$  sums to 1 , each element of  $\tilde{A}^t x$  is an convex combination of the components of  $x$  , which can't be greater than a  $x_{max}$  where  $x_{max}$  is the maximum element of the vector  $x$  .

Let's see it formally

$$x_{max} := \max_{j \in \mathbb{N} \cap [1, n]} |x_j|$$

$$\alpha := \tilde{A}^t$$

$$\forall \mathbb{N} \cap [1, n] \sum_{j=1}^n \alpha_{ij} = 1$$

$$\begin{aligned} \forall x \in \mathbb{R}^n, \forall i \in \mathbb{N} \cap [1, n], \sum_{j=1}^n \alpha_{ij} x_j &\leq \sum_{j=1}^n \alpha_{ij} x_{max} = \\ &= x_{max} \sum_{j=1}^n \alpha_{ij} = x_{max} \end{aligned}$$

In conclusion

$$(\tilde{A}^t x)_{max} \leq x_{max} \quad (2)$$

On the other hand , At least one element of  $\lambda x$  is greater than  $x_{max}$  , which proves that  $\lambda > 1$  is impossible

Or , more precisely , let  $\lambda > 1$  be an eigen value of the matrix  $\tilde{A}^t$  and let  $v_\lambda$  be the normalized eigen vector of eigen value  $\lambda$  then

$$(v_\lambda)_{max} < \lambda(v_\lambda)_{max} = (\lambda v_\lambda)_{max} = (\tilde{A}^t v_\lambda)_{max} \quad (3)$$

By equation 2 we can select  $x := v_\lambda$  and we'll have  $(\tilde{A}^t v_\lambda)_{max} \leq (v_\lambda)_{max}$  which contradict equation 3

## Introduction

## TODO Simple pseudo code AT-TACH

- Note taken on *[2023-07-05 Wed 14:42]*  
Hello World !!!
- Note taken on *[2023-07-05 Wed 14:37]*  
BARAKKKK
- Note taken on *[2023-07-05 Wed 14:36]*  
Hello World !!
- Note taken on *[2023-07-05 Wed 14:36]*  
Barak the King

Hello World

$\lambda * 6$

0:00:23 0:01:17 0:01:20 0:01:22 The Org Manual  
0:01:22

0:01:22 0:01:22 0:01:22 0:01:22 0:01:22 0:01:22  
0:01:22 0:01:22 0:00:03 0:00:05 0:00:07 0:00:08 0:00:09  
0:00:09 0:00:00 0:00:03 0:00:04 0:00:06 0:00:07 0:00:00  
0:00:00 0:00:02 0:00:03 0:00:02 0:00:03

IDEA

KILL

IDEA