Centrality in Graphs

Atiyeh Sayadi

February 5, 2024

Contents

Revision History

Date	Version	Notes
January 29, 2024	1.0	Initial drafts
Feburary 4, 2024	1.0	First review

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

In this project there is no special unit.

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
DC(i)	–	Degree Centrality of node i
CC(i)	–	Closeness Centrality of node i which heat is transferred in

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
CIG	Centrality in Graphs: finding the most importamt nodes
TM	Theoretical Model

1.4 Mathematical Notation

There is no mathematical notation.

2 Introduction

Analyzing networks using graph-related concepts has been one of the most interesting areas of study for professionals. Among these concepts, centrality and identifying the most effective members of the network are crucial for some studies. This project attempted to identify the most important nodes in a given graph, represented in the form of a matrix of nodes and relationships, using degree centrality and closeness centrality methods. In fact, our goal is to calculate the centrality measures of each node in the graph using both of the mentioned methods.

2.1 Purpose of Document

The purpose of this document is to provide an overview of the design and implementation project for a system to compute centrality and degree in graphs. Therefore, using this document, various dimensions of the project including requirements, objectives, implementation methods, etc., will be clearer, which aids in the development and maintenance of the software

2.2 Scope of Requirements

This project does not encompass various methods for calculating centrality in graphs. In other words, it only utilizes the closeness and degree methods. Additionally, selecting the best criterion for centrality computation is not within the scope of this project's implementation agenda. Lastly, weighted and directed graphs will not be examined in this project.

2.3 Characteristics of Intended Reader

The intended readers for this project are individuals with sufficient knowledge in graph theory and network analysis, in addition to those with programming expertise.

2.4 Organization of Document

his document follows the structure proposed by Smith and Lai (2005), covering all opportunities, requirements, definitions, etc. So, readers can start by looking at the goal statements and then dive into the theoretical model to fully understand the system.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Looking at Figure 1 gives you a basic idea of the system. First, it reads a graph file or text file with a graph. Then, it calculates closeness measures for each node using centrality and closeness definitions, giving two arrays as output for the given graph.

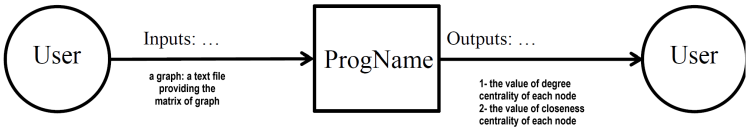
User
Inputs: ...
a graph: a text file providing the matrix of graph
ProgName
Outputs: ...
1- the value of degree centrality of each node
2- the value of closeness centrality of each node
User

Figure 1: System Context

- User Responsibilities:
  - the matrix of arrey should be normalized.
- CIG Responsibilities:
  - read the input file
  - calculating degree and closeness centrality for each node in the graph correctly

3.2 User Characteristics

Users of this system need to know about networks and graphs to understand the output based on their input. They should also be able to upload different files if needed.

3.3 System Constraints

This system does not examine weighted or directed graphs. Additionally, it does not encompass all types of centrality calculation methods.

4 Specific System Description

Consider having a graph representing a social network. We aim to identify the most central individuals in this network using the concept of centrality. Firstly, the social network needs to be introduced to the system in the form of a graph. Then, the software calculates the centrality measures, specifically degree centrality and closeness centrality, for each node individually and ultimately computes them for all members of the network.

4.1 Problem Description

We have a graph representing a social network, and we aim to identify the most central individuals in this network using the concept of centrality. In other words, finding the most important members in a network in terms of their connections with other members is the main objective of this project.

4.1.1 Terminology and Definitions

- **Graph:** A collection of nodes connected by edges.
- **Node:** Symbole of members in graphs.
- **Edghe:** Connections between nodes in graphs.
- **Centrality:** Measure used to identify the most important nodes in a network.
- **Degree:** Number of direct connection that each node has.

4.1.2 Physical System Description

This project can model any network in which directionality is not significant, in the form of an undirected graph. In this network, it is preferred that there are no isolated nodes, although their presence does not disrupt computations, it increases the complexity of analysis. In fact, all networks with bidirectional communication paths can be modeled by this system.

4.1.3 Goal Statements

GS1: Calculating degree centrality for each node.

GS2: Calculating closeness centrality for each node.

4.2 Solution Characteristics Specification

As previously mentioned, the solution presented in this system is only applicable to networks where the direction of communication between members is not significant, in other words, bidirectional relationships. Additionally, networks where communications do not have varying weights or values can be read as input in this system.

4.2.1 Assumptions

To simplify the calculations, some assumptions have been made in this software:

A1: The graph is undirected.

A2: The length of each edge is equal to 1.

4.2.2 Theoretical Models

The purpose of this project is to calculate centrality and proximity measures in a graph. Therefore, it is necessary to become more familiar with this concept accurately.

RefName: **TM:DC**

Label: Degree Centrality

Equation: $DC(i) = \frac{\sum \text{edge}(i,j)}{N-1}$

Description: Degree centrality looks at how many connections each node has, and if a node has a lot of connections, it's considered more important.to calcute DC for each node we need at first calculate degree of node. Degree of node is the number of edged that are direcly connectod to the node. After that we should devide the result to the number of nodes in the graph -1.

Notes: None.

Source: https://www.geeksforgeeks.org/degree-centrality-centrality-measure/

Ref. By: GD3,IM1

Preconditions for TM:DC: None

Derivation for TM:DC: Not Applicable

RefName: TM:CC

Label: Closeness Centrality

Equation: $CC(i) = \frac{N-1}{\sum d(i,j)}$
d(i,j) = The shortest path from i to j.

Description: It checks how close a node is to all the other nodes in the graph. To calculate average distance from node i to all other node, the sum of shortest path from node i to all other nodes should be computed, and the result must divide to total number of nodes -1. Then it must be reversed to find the CC node i.

Notes: None.

Source: https://en.wikipedia.org/wiki/Centrality

Ref. By: GD4,IM2

Preconditions for TM:CC: None

Derivation for TM:CC: Not Applicable

4.2.3 General Definitions

In this section you can finde some general about graph, centrality, degree centrality and closeness centrality.

Number	GD1
Label	**Graph**
SI Units	–
Equisions	–
Description	Graph is a collection of nodes connected by edges. It has applications in many fields such as algorithm design and analysis, game theory and network theory.
Source	https://www.geeksforgeeks.org/graph-types-and-applications/
Ref. By	TM1, TM2,GD2,GD3,GD4,DD1,IM1,IM2

Number	GD2
Label	**Centrality**
SI Units	–
Equisions	–
Description	”Centrality” is a measure used to identify the most important nodes in a network.
Source	https://visiblenetworklabs.com/2021/04/16/understanding-network-centrality/
Ref. By	TM1, TM2,GD3,GD4,IM1,IM2

Number	GD3
Label	**Degree Centrality**
SI Units	–
Equisions	$DC(i) = \frac{\sum \text{edge}(i,j)}{N-1}$
Description	Degree centrality looks at how many connections each node has, and if a node has a lot of connections, it’s considered more important.
Source	https://www.geeksforgeeks.org/degree-centrality-centrality-measure/
Ref. By	TM1, IM1

Number	GD4
Label	**Centrality Centrality**
SI Units	–
Equisions	$CC(i) = \frac{N-1}{\sum d(i,j)}$ d(i,j) = The shortest path from i to j.
Description	Checks how close a node is to all the other nodes in the graph.
Source	https://en.wikipedia.org/wiki/Centrality
Ref. By	TM2, IM2

4.2.4 Data Definitions

In this section, a general definition of the degree of node, which is highly important in graph theory, is presented.

Number	DD1
Label	**The Degree of Node**
Symbol	–
SI Units	
Equation	Degree(i)=$\sum edge(i,j)$
Description	In graph theory, the degree of a node is the number of edges directly connected to it.
Sources	https://www.geeksforgeeks.org/degree-centrality-centrality-measure/
Ref. By	TM1,GD1,GD2,GD3,DD1,IM1

4.2.5 Data Types

In this project, the input is a matrix of natural numbers representing nodes and edges, and the output is a set of floating-point numbers indicating the centrality of each node in the graph.

4.2.6 Instance Models

In this section, a computational model of degree and closeness centrality for each node of the given graph is presented.

Number	IM1
Label	Degree Centrality
Input	The matirix of the graph
Output	The value of degree centrality for each node
Description	it shows how many direct connection have the the nodes in the graph.
Sources	https://en.wikipedia.org/wiki/Degree_(graph_theory)
Ref. By	-

Number	IM2
Label	Closeness Centrality
Input	The matirix of the graph
Output	The value of closeness centrality for each node
Description	it shows how close the nodes are to others.
Sources	https://en.wikipedia.org/wiki/Centrality)
Ref. By	-

4.2.7 Input Data Constraints

The input utilizes a text file to present the matrix of relationships in the graph. In this matrix, the elements are integers greater than zero.

Table 1: Input Variables

Var	**Physical Constraints**	**Software Constraints**	**Typical Value**	**Uncertainty**
$MatrixM$	$\lvert M \rvert > 0$	Integer value	1	-

4.2.8 Properties of a Correct Solution

Output of this software is a set of numbers representing the centrality of each node in both methods. This value for each method ranges between zero and one.

Table 2: Output Variables

Var	**Physical Constraints**
$DC(i)$	$0 <= DC(i) <= 1$
$CC(i)$	$0 <= CD(i) <= 1$

5 Requirements

This section delineates the requirements of this project across various sections.

5.1 Functional Requirements

R1: A text file containing the adjacency matrix of a directed and unweighted graph.

R2: The computational formulas must be accurately implemented by the software.

R3: A set of numerical values representing the centrality of each node calculated using both degree centrality and closeness centrality methods. These values range between zero and one.

R4: A set of numerical values representing the centrality of each node calculated using both degree centrality and closeness centrality methods. These values range between zero and one.

5.2 Nonfunctional Requirements

In this section, non-functional requirements of the software are expressed in the form of various criteria.

NFR1: **Accuracy** Efforts must have been made to accurately implement computational algorithms to ensure reliable and correct output.

NFR2: **Usability** Codes should be written in an invironment that it is easy to users to run, like python.

NFR3: **Maintainability** Efforts should be made to minimize side effects between functions at minimum. Additionally, it is necessary for the names of functions, variables, and parameters to clearly reflect their functionality.

NFR4: **Portability**

This software must be written in a language that is capable of running in any environment with any operating system.

5.3 Rationale

In this project, efforts have been made to avoid using weighted and directed graphs because these graph models increase computational complexity. Additionally, since the degree centrality and closeness centrality calculations yield different results in directed graphs, the computation methods would also differ.

6 Likely Changes

LC1: In this project, it is possible that the programming environment may change.

LC2: Additionally, the method of defining the matrix in the input file may change from n*2 to n*n.

7 Unlikely Changes

LC3: The input file is a text file.

LC4: Only two centrality methods, degree centrality and closeness centrality, are used.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 3 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 4 shows the dependencies of instance models, requirements, and data constraints on each other. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	TM1	TM2	GD1	GD2	GD3	GD4	DD1	IM1	IM2
TM1					X			X	
TM2						X			X
GD1	X	X	X	X	X	X	X	X	X
GD2						X	X	X	X
GD3								X	
GD4									X
DD1	X		X	X	X			X	
IM1									
IM2									

Table 3: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2
R1	X	X
R2	X	X
R3	X	X
R4	X	X

Table 4: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A1	A2
TM1	X	
TM2	X	X
GD1		
GD2		
GD3	X	
GD4	X	X
DD1	X	
IM1	X	
IM2	X	X

Table 5: Traceability Matrix Showing the Connections Between Assumptions and Other Items

9 Development Plan

For software development, first, the input file needs to be standardized and read from the input. Then, two centrality algorithms, degree centrality and closeness centrality, are implemented, and efforts are made to display the output with relying on computations and definitions of other matrices.

10 Values of Auxiliary Constants

Input matrix:: as an input we have a matrix[n:2]. It means it has several rows and just two columns.

References

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

.

Newman. (2013). Networks: An Introduction.

Wasserman, S. and Faust, K. (1994). Social Network Analysis: Methods and Applications.

https://www.geeksforgeeks.org/degree-centrality-centrality-measure/

https://en.wikipedia.org/wiki/Centrality
https://visiblenetworklabs.com/2021/04/16/understanding-networkcentrality/

https://www.geeksforgeeks.org/graph-types-and-applications/

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

2. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

3. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?