Birla Institute of Technology and Science, Pilani – Hyderabad campus

A Project Report on

Visibility Graph Networks

SUBMITTED AS AN EVALUATION COMPONENT OF

Research Practice BITS G540

Submitted by,
Ayush Jain (2020H1240079H)
YEAR 1, SEM II

Under the guidance of
Dr. Rajesh Kumar Tripathy
Assistant Professor
Department of Electrical and Electronics Engineering

Submission Date,
9th June, 2021

Visibility Graph Networks

Horizontal Visibility Graph:

Visibility Graph algorithm helps to map the time series to complex networks. Each sample of a time series can be considered as a node of the graph as represented in a histogram. The value corresponding to the data node is represented by the height of the histogram. Two nodes will be connected to each other if the top of two bars is visible. Edge between the two nodes (t_i, x_i) and (t_j, x_j) will be connected to each other if any node (t_k, x_k) between (t_i, x_i) and (t_j, x_j) fulfils this condition:

$$\frac{x_i - x_k}{t_k - t_i} > \frac{x_i - x_j}{t_j - t_i}, t_i < t_k < t_j, \quad (1)$$

Horizontal Visibility Graph algorithm is a modification in Visibility Graph algorithm. There is horizontal visibility between the two nodes (t_i, x_i) and (t_j, x_j) if they fulfils the following condition:

$$x_i, x_j > x_k, \; t_i < t_k < t_j, \quad (2)$$

where, (t_k, x_k) is the data node between (t_i, x_i) and (t_j, x_j).

Note: *A* ***time series*** *is a set of observation taken at specified times usually at equal intervals. It is used to predict the future values based on the previous observed values.*

Steps followed to create HVG:

1. Firstly the time series data is taken as given in the paper:
   series = [7.0, 4.0, 8.0, 6.5, 7.6, 9.0]
2. Then, the histogram is plotted as shown in the figure given below for the given time series and nodes are represented.

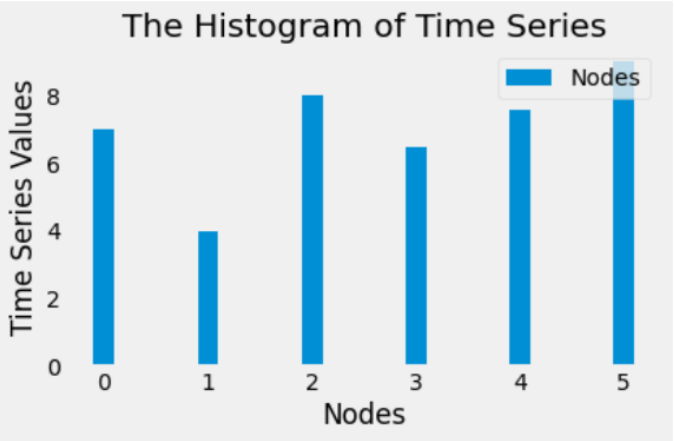

Fig. 1 The Histogram of Time Series

3. Now, to create its corresponding HVG, the complex network can be represented as adjacent matrix $A = (a_{ij})_{NxN}$. If t_i and t_j are connected, then, $a_{ij} = 1$, otherwise 0.

   Algorithm to create adjacent matrix:

   Step 1: Start

   Step 2: Convert time series into numpy array

   Step 3: Create a numpy array for nodes

   Step 4: Initialize a 2-D List

   Step 5: Set a loop up to row

   - Set an inner loop up to the column
   - Store the magnitude difference in a variable
   - if the value of variable is zero
   - Set the edge weight between the two nodes equal to 0
   - elif the value of variable is one
   - Set the edge weight between the two nodes equal to 1
   - Else check if there is any node whose value is greater than these two
   - If yes then set the weight between the two nodes equal to 0
   - If no then set the weight between the two nodes equal to 1

   Step 6: Convert the 2-D List to numpy array

   Step 7: Print the Adjacent Matrix

   Step 8: Stop

4. Obtained Adjacent Matrix is:

```
[[0 1 1 0 0 0]
 [1 0 1 0 0 0]
 [1 1 0 1 1 1]
 [0 0 1 0 1 0]
 [0 0 1 1 0 1]
 [0 0 1 0 1 0]]
```

5. Create a function for plotting the HVG graph.
6. Plotted the graph using Networkx and Matplotlib libraries. Obtained graph is shown in the figure below:

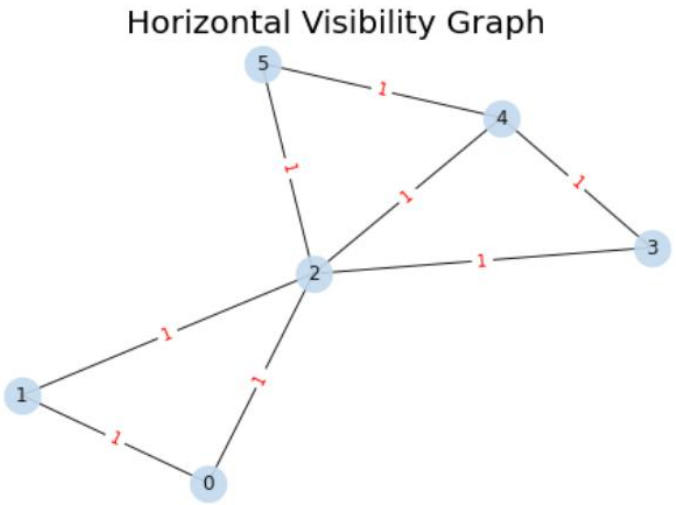

Fig. 2 Horizontal Visibility Graph

Directed Weighted Horizontal Visibility Graph:

In this the edge weight is related to the visibility angle measurement and the weighted complex network can be represented by weight matrix $W = (w_{ij})_{NxN}$. It is of two types:

i. Forward Weighted Horizontal Visibility Graph (FWHVG)

In this the angle between the nodes is calculated by using the formula:

$$w_{ij}^{f} = \arctan\frac{x_j - x_i}{t_j - t_i} + \arctan\frac{x_i}{t_j - t_i}, t_i < t_j, \quad (3)$$

ii. Backard Weighted Horizontal Visibility Graph (BWHVG)

In this the angle between the nodes is calculated by using the formula:

$$w_{ij}^{b} = \arctan\frac{x_i - x_j}{t_j - t_i} + \arctan\frac{x_j}{t_j - t_i}, t_i < t_j \quad (4)$$

Steps followed to create FWHVG:

1. Firstly the time series data is taken as given in the paper:

   series = [7.0, 4.0, 8.0, 6.5, 7.6, 9.0]

2. Then, the histogram is plotted as shown in the figure given below for the given time series and nodes are represented. Angle is measured as shown below.

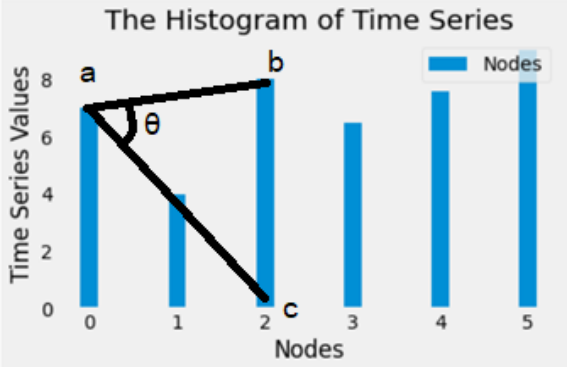

Fig. 3 Angle measurement of FWHVG

3. Now, to create its corresponding FWHVG, the complex network can be represented as weight matrix $W = (w_{ij})_{NxN}$. If t_i and t_j are connected, then, a_{ij} = angle between the nodes i and j, otherwise 0.

   Algorithm to create weight matrix for FWHVG:

   Step 1: Start

   Step 2: Convert time series into numpy array

   Step 3: Create a numpy array for nodes

   Step 4: Initialize a 2-D List

   Step 5: Set a loop up to row

   - Set an inner loop up to the column
   - Store the magnitude difference in a variable
   - if the value of variable is zero
   - Set the edge weight between the two nodes equal to 0
   - Elif the row is less than the column to measure the angle in forward direction
     - If the difference between the nodes i.e. value of variable is equal to one
     - Set the edge weight between the two nodes equal to angle between them
     - Else check if there is any node whose value is greater than these two
     - If yes then set the weight between the two nodes equal to 0
     - If no then set the weight between the two nodes equal to angle between

them

- Else set the weight between the nodes equal to 0.

Step 6: Convert the 2-D List to numpy array

Step 7: Print the Adjacent Matrix

Step 8: Stop

4. Obtained Weight Matrix is:

```
[[0.    0.18  1.756 0.    0.    0.   ]
 [0.    0.    2.652 0.    0.    0.   ]
 [0.    0.    0.    0.464 1.128 1.534]
 [0.    0.    0.    0.    2.251 0.   ]
 [0.    0.    0.    0.    0.    2.391]
 [0.    0.    0.    0.    0.    0.   ]]
```

5. Create a function for plotting the FWHVG graph.

6. Plotted the graph using Networkx and Matplotlib libraries. Obtained graph is shown in the figure below:

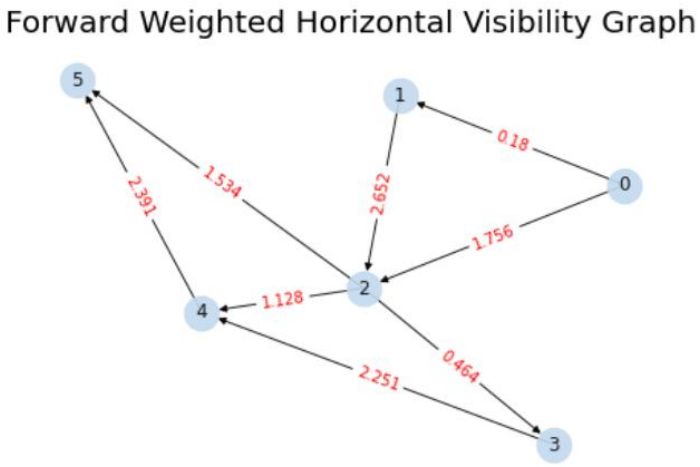

Fig. 4 Forward Weighted Horizontal Visibility Graph

CONCLUSION:

We have successfully created Horizontal Visibility graph and Forward Weighted Horizontal Visibility graph for a time series data and results are matching with the paper. In future we will use FWHVG and BWHVG to extract network features and fuse these two into a single feature matrix to classify EEG signals.

REFERENCES:

[1] Kong, Tianjiao & Shao, Jie & Hu, Jiuyuan & Yang, Xin & Yang, Shiyiling & Malekian, Reza. (2021). EEG-Based Emotion Recognition Using an Improved Weighted Horizontal Visibility Graph. Sensors. 21. 1870. 10.3390/s21051870.

[2] S. Yan and D. Wang, "Time Series Analysis Based on Visibility Graph Theory," *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2015, pp. 311-314, doi: 10.1109/IHMSC.2015.238.