Comprehensive Evaluation Method for Network Node Importance based on Constructed Tree

Yang Yuxin *, Peng Yali a, Huang Minghe b
School of Software, Jiangxi Normal University, Nanchang, 330022

* 939101457@qq.com, 16606073616
a 3494534283@qq.com, 13453298800, b 1152353630@qq.com, 13170865915

ABSTRACT The importance measurement of nodes in directed weighted networks is of great significance for theoretical research and practical application of networks. Aiming at the directed weighted network model in actual networks, based on the special transformation of graphs and tree roots, this paper proposes a comprehensive quantitative assessment method for the importance index of nodes based on the directed weighted network's global properties, local attributes, network location and bidirectionally propagation. This method transforms the nodes to be analyzed in the graph to root, constructs a bidirectional depth-affected XD tree and a breadth-affected XB tree with the node to be evaluated as the root of the tree. Then comprehensively consider the two-way influence of the node level, node neighbors and the closeness between neighbors in the two trees constructed, and the in-out effects, build a deep impact evaluation model at the two-way level, two-way neighbor information and comprehensive evaluation index of clustering coefficient. By calculating the comprehensive importance value of each node in the network, the comprehensive ranking of the importance of the nodes in the network can finally be obtained. The robustness experiment of ARPA network shows that the G value of the important node quantified by this method has absolute advantages, this method can effectively analyze key nodes in the network.

1. INTRODUCTION
Complex networks have heterogeneous topologies, which determine the varied importance of nodes in the network. Locating the key nodes in the network accurately will be helpful to control the network and improve the reliability and invulnerability of the network. For example, when rumor is spreading on network, if one can accurately locate the key nodes in the network and handle the rumor from there, then the spread of rumors can be quickly and effectively controlled. And when virus is propagating in the Network, accurately identifying the key nodes has great practical importance. Same situations are to be found in Network like literature reference and power networks.

In real life, there are a large number of networks is directed weighted network, like public bicycle dispatching system. The scheduling of bike includes two part, the inter-network scheduling, and the number of scheduling. However, current studies on the key nodes of complex network focus mainly on undirected or unweighted network, and there are few studies on directed weighted networks. While the study of the importance of nodes in the directed weighted network can improve practical value of the research on complex networks.

Complex networks have scale-free and small-world characteristics, so the failure of important nodes in the network may lead to the collapse of the entire network, affecting the overall situation. At present, the metrics for key nodes querying in directed weighted networks include degree centrality, PageRank,
eigenvector centrality, eccentricity, closeness centrality, betweenness centrality, clustering coefficient and so on. The assessment method based on the importance of degree emphasizes on the number of edges between node and its adjacent nodes, hence, it can reflect the importance of nodes in the network to some extent, but nodes with the same degree are often of different importance in the network; betweenness describes the control of the nodes or edges on the information flow in the network, however, the betweenness centrality is generally defined according to the shortest path which is not case in the actual networks --- information flow according to the willingness. Therefore, in some networks, the betweenness centrality is not applicable to the measure of the importance of nodes; At present, measurement algorithms for the key nodes in directed weighted networks include DWCN-NodeRank based on the PageRank algorithm, but DWCN-NodeRank can not provide good evaluation precision and fast convergence speed at the same time; For a directed network, most of the existing methods only consider the neighbor nodes of the evaluated node, while the influence of the evaluated node’s non-neighbor nodes are neglected. However, in the network with high density, the influence of non-neighbor nodes can not be neglected, as it will lead to the inaccuracy of evaluation conclusion.

Based on the directed weighting network, this paper proposes to abstract the original map so as to construct a deep traversal tree consist of nodes to be evaluated. By studying influence of its roots, influences of different nodes on the root node can be fully evaluated; Since the influence of some nodes directly connected to the node to be evaluated is neglected in the process of construct tree, we use the normalized clustering coefficient to reflect the influence of all neighbor nodes on the evaluated node. Through a reasonable combination of the two, the importance of different nodes in the network can be accurately reflected.

2. XD TREE SYSTEM AND XB TREE SYSTEM CONSTRUCTION METHOD BASED ON NETWORK DIAGRAM

2.1 XD tree system construction method
According to actual requirements, the node to be evaluated in the graph is designated as the root node. Starts a deep traversal based on the specified root node. The depth traversal of the graph is slightly more complicated than the depth traversal of the tree.

The basic idea of depth-first traversal of the tree structure is: start from the root node V0 of the tree, visit V0, then select the unvisited node Vi from the child nodes of V0, and then select a non-visited child node from Vi. The visited vertices Vj are visited and continue in order. If all child vertices of the currently visited vertex have been visited, then return to the node with the most recently visited child nodes and traverse in the same way until all the vertices in the tree have been visited.

For deep traversal of the graph, you need to mark all the nodes in the graph as unvisited, then select a source node, mark it as visited, and then recursively use the depth-first search method to sequentially search all adjacent points of the source node , if there are unvisited adjacencies, the new adjacency discovered is used as the origin to continue the traversal. When all the nodes with a path from the source node are visited, the traversal for the selected source node ends.

The order of access to nodes by deep traversal is related to the storage state of the adjacency list of the graph. The adjacency list of the graph is not unique, so the depth traversal results of the same graph may also be different. The graph’s depth traversal algorithm is as follows:

```c
void DFS(LinkedGraph g, int i){
    EdgeNode *p; printf("visit vertex:%c \n", g.adjlist[i].vertex);
    visited[i] = 1; p = g.adjlist[i].FirstEdge;
    /* Depth-first traversal from the adjacent points of p */
    while (p){ if(!visited[p->adjvex])
        dfs(g,p->adjvex); p = p->next; } }
```
The function of the above algorithm is: from the vertex i to the connected components of the depth-first traversal graph, where g represents the adjacency list of the graph, i represents the starting point of the traversal, and this function has no return value.

```c
void DfsTraverse(LinkedGraph g){
    int i;
    for(i=0;i<g.n;i++)
        visited[i] = 0;     /* Initialize the flags array */
    for(i=0;i<g.n;i++)
        if(!visited[i])
            dfs(g,i);
/* vi has not been visited */
}
```

The function of the above function is: depth-first traversal of the graph, g represents the adjacency list of the graph, this function has no return value.

For a graph with n nodes and e edges, the traversal algorithm DfsTraverse calls the DFS function for each node in the graph at most once. The maximum number of times to call DFS from DfsTraverse function or internally recursively call DFS function is n. Therefore, once all nodes are accessed, all nodes in the edge table need to be checked once on the adjacency list, so the time complexity of the DfsTraverse algorithm is \( O(n + e) \).

Through the above method, a deep traversal tree is obtained with the node to be evaluated as the root node and the path between nodes in the original image as the continuous edge.

After the first deep traversal tree is obtained, not all nodes in the graph may necessarily be traversed. If there are still nodes that are not traversed in the graph, at this time, you can continue to specify the next node to be evaluated as the root node among all the nodes that have not been traversed, and perform a second deep traversal among all the nodes that have not been traversed. Get the second depth traversal tree, continue to specify the root node and perform the depth traversal until all nodes in the graph have been traversed, and the traversal ends. All the above processes, starting from the first designated root node and until all nodes in the graph are accessed, all generated trees are called the XD forest of the first designated root node.

### 2.2 XB tree system construction method

According to the actual requirements, the node to be evaluated in the graph is designated as the root node, and the breadth traversal is started based on the specified root node. The breadth-first traversal idea of the graph is roughly the same as the tree level traversal idea.

Breadth-first traversal of a tree is similar to hierarchical traversal. The basic idea of breadth-first traversal of a tree is: starting from the root node of the tree, starting to access the nodes in the tree; Each of the unvisited child nodes of the root node is visited in turn; after that, starting from the visited node, all of the child nodes that have not been visited are visited in turn.

The search algorithm adopted by the breadth-first traversal method of the graph is characterized by searching horizontal nodes first as much as possible. The basic idea is: starting from a certain source node in the graph, after visiting the source point, search all adjacent nodes of the source node as horizontally as possible. After successively accessing the neighboring nodes of the source node, starting from these neighboring points, the neighboring nodes that are not visited by each node are sequentially visited.

Adopt the adjacency list storage structure for the graph. The breadth-first traversal algorithm is as follows:

```c
void BFS(LinkedGraph g,int i){
    int j; EdgeNode *p; int queue[M],front,rear;
    front = rear = 0;       /* Initialize the empty queue */
    printf("%c",g.adjlist[i].vertex); /* Visit source i */
    visited[i] = 1; queue[rear++] = i;       /* Enqueue the visited node */
/* When the queue is not empty, execute the following loop */
    while(rear>front){
```
The function of the above function is: traverse the connected components of the graph g by breadth first from the vertex i; the parameter g in the function represents the adjacency list of the graph, i represents the traversal of the source points; the function has no return value.

```c
int BfsTraverse(LinkedGraph g){
    int i, count = 0;
    for(i = 0; i < g.n; i++)
    {
        visited[i] = 0;  /* Initialize the flags array */
        for(;; i++)
        {
            if(!visited[i])
            {
                /*vi is not visited */
                printf("\n"); count++;
                BFS(g, i);    } return count;
        }
    }
}
```

The function of the above function is: breadth-first traversal of graph g. The parameter g in the function indicates the adjacency list. The function has a return value, and the return value is the number of connected components.

For a graph with n nodes and e edges, each node is enqueued once. The time complexity of the algorithm BfsTraverse is the same as the DfsTraverse algorithm, and it is also O (n + e). The number of times the BFS function is called during the breadth traversal of the soil is the number of connected components in the graph.

Through the above method, a breadth traversal tree with the node to be evaluated as the root node and the path between nodes in the original image as the connected edges is obtained.

After the first breadth traversal tree is obtained, it is the same as the XD forest, and all nodes in the graph may not necessarily be traversed. If there are still nodes that are not traversed in the graph, you can continue to designate the next node to be evaluated as the root node among all the nodes that have not been traversed. Perform further breadth traversal among all the nodes that have not been traversed to get the second, third ... deep traversal tree, continuously specify the root node and perform deep traversal until all nodes in the graph have been traversed, and the traversal ends. All the above processes, starting from the first designated root node and until all nodes in the graph are visited, all generated trees are called the XB forest of the first designated root node.

3. NODE IMPORTANCE EVALUATION METHOD BASED ON DEPTH INFLUENCE

3.1 Deep impact assessment method
In social networks, the three-degree theory means that our own actions can be applied to our friends, relatives or colleagues, as well as to their friends, relatives or colleagues. It can also act on friends, relatives or colleagues who indirectly relate to friends, but the force between them is gradually weakened. Based on the third degree theory, the importance of the nodes in the network graph is not only related to the nodes that were once connected to them, but also related to other nodes radiated in the network. That is to say, in the network to be studied, some nodes that are not directly connected to the node to be evaluated will have an indirect influence on the evaluation of the importance of the network node. We record this influence as: p.
Due to different practical application problems, the degree of ingress and egress of the root node may have different degrees of influence on the importance of the root node. Therefore, the indirect influence formula is expressed as:

\[ P_i = \alpha P_{in} + (1 - \alpha) P_{out}. \]

According to the third degree theory, nodes at different levels in the tree decrease as the distance from the root node increases. Taking \( P_{out} \) as an example, if the number of nodes in the entire network to be evaluated is \( n \), in the tree with the node to be evaluated as the root node, the sum of the degrees of connection between all nodes whose depth is equal to 1 and their parent nodes is \( d_{i1} \). The sum of the degrees of connection between nodes with depths equal to 2 and their respective parent nodes is \( d_{i2} \)... So on and so forth, the depth influence coefficient is \( \lambda \), and the value should be greater than 0 and less than or equal to 1. \( N_t \) represents the set of fixed points that point \( i \) is reachable in step \( t \). Then, the formula of depth influence based on the degree tree is expressed as:

\[
P_{out} = \frac{1}{n-1} \left( \lambda \sum_{i \in N_1} d_i + \lambda^2 \sum_{i \in N_2} d_i + \lambda^3 \sum_{i \in N_3} d_i + \cdots + \lambda^n \sum_{i \in N_n} d_i \right)
\]

3.2 Breadth impact assessment method

The evaluation of the depth influence index based on the XD tree fully reflects the degree of influence of different depth nodes on the root node, that is, the importance of the node in the entire network. However, in the process of importance evaluation based on the XD tree, the degree of influence of the nodes directly connected to the root node on the nodes to be evaluated may be weakened in some original networks, that is, the regional importance of the nodes within a certain range. The degree of influence of once connected nodes on the importance of the root node is much greater than that of other deep nodes on the root node. In the figure, the clustering coefficient can reflect the proportion of neighbor nodes that are neighbors to each other. A tree is a special case of a graph. All the evaluation indicators of the graph can be used in the tree. Here, the clustering coefficient index is used to evaluate the influence of neighboring nodes in the XB tree to be evaluated. We use \( Q_i \) to represent the influence of node breadth.

The main process of this method is: Assume that the network \( G = (V;E) \) is an undirected network composed of \( |V| = N \) nodes and \( |E| = M \) edges. The degree indicator describes the number of neighbor nodes of a node, expressed as \( K_i = \sum_{j \in E} \partial_{ij} \) indicates whether i and j have two sides. If there is a side \( \partial_{ij} \), it is 1, otherwise it is 0. The degree indicator reflects the ability of the node to establish a direct connection with the surrounding nodes, but does not reflect the connected side of the node's neighbor nodes. The aggregation coefficient \( c_i \) describes the proportion of neighbors of nodes in the network that are neighbors, \( e_i \) represents the number of triangles formed between node i and any two neighbor nodes. Therefore, the agglomeration coefficient \( c_i \) is expressed as:

\[
Q_{in} = \frac{2e_i}{k_i(k_i - 1)}
\]

The degree of influence of the degree of influence of the node to be evaluated \( Q_{in} \) is the same as that of \( q_{out} \). The direct influence of the root node, that is, the breadth influence is expressed as:

\[
Q_i = \beta Q_{in} + (1 - \beta) Q_{out}.
\]

3.3 Comprehensive evaluation method of deep influence and breadth influence

The importance of outlets is affected by both depth influence and breadth influence. In practical applications, the two can be given different influence coefficients \( \lambda \). Therefore, the calculation formula of the final influence evaluation index \( K_i \) is obtained.

\[
K_i = \lambda P_i + (1 - \lambda) Q_i
\]
4. CASE ANALYSIS

The ARPA network is a network that is commonly used when analyzing the importance of network nodes. The network has 21 nodes and 26 edges. Here we calculate and evaluate the importance of each node in the ARPA network based on the network node importance evaluation method described in this article. Because the calculation method of depth influence is tedious, the calculation process is listed in the following table.

Table 1 Demonstration of calculation of deep influence value in ARPA network

| node | ki | αi | βi | pi | ki | αi | βi | pi | pi |
|------|----|----|----|----|----|----|----|----|----|
| 1    | 3  | 0.15 | 0.068 | 0.218 | —  | —  | —  | —  | —  | 0.218 |
| 2    | —  | —  | —  | —  | 8  | 0.4 | 0.1952 | 0.5952 | 0.5952 |
| 3    | 2  | 0.1 | 0.02 | 0.12 | 4  | 0.2 | 0.088 | 0.288 | 0.288 |
| 4    | 2  | 0.1 | 0.048 | 0.148 | 1  | 0.05 | 0.02 | 0.07 | 0.148 |
| 5    | 4  | 0.2 | 0.0992 | 0.2992 | —  | —  | —  | —  | —  | 0.2992 |
| 6    | —  | —  | —  | —  | 5  | 0.25 | 0.1192 | 0.3692 | 0.3692 |
| 7    | 1  | 0.05 | 0.02 | 0.07 | 2  | 0.1 | 0.048 | 0.148 | 0.148 |
| 8    | 2  | 0.1 | 0.048 | 0.148 | 1  | 0.05 | 0.02 | 0.07 | 0.148 |
| 9    | 7  | 0.35 | 0.1909 | 0.5409 | —  | —  | —  | —  | —  | 0.5409 |
| 10   | 3  | 0.15 | 0.0792 | 0.2292 | 1  | 0.05 | 0.02 | 0.07 | 0.2292 |
| 11   | 2  | 0.1 | 0.048 | 0.148 | 2  | 0.1 | 0.048 | 0.148 | 0.148 |
| 12   | 1  | 0.05 | 0.02 | 0.07 | 4  | 0.2 | 0.0992 | 0.2992 | 0.2992 |
| 13   | 3  | 0.15 | 0.068 | 0.218 | —  | —  | —  | —  | —  | 0.218 |
| 14   | —  | —  | —  | —  | 5  | 0.25 | 0.108 | 0.358 | 0.358 |
| 15   | 2  | 0.1 | 0.04 | 0.14 | 1  | 0.05 | 0.02 | 0.07 | 0.14 |
| 16   | 2  | 0.1 | 0.04 | 0.14 | —  | —  | —  | —  | —  | 0.14 |
| 17   | 3  | 0.15 | 0.068 | 0.218 | —  | —  | —  | —  | —  | 0.218 |
| 18   | 3  | 0.15 | 0.068 | 0.218 | —  | —  | —  | —  | —  | 0.218 |
| 19   | —  | —  | —  | —  | 8  | 0.4 | 0.2077 | 0.6077 | 0.6077 |
| 20   | 1  | 0.05 | 0.02 | 0.07 | 1  | 0.05 | 0.02 | 0.07 | 0.07 |
| 21   | 3  | 0.15 | 0.068 | 0.218 | —  | —  | —  | —  | —  | 0.218 |

Here we analyze the robustness of the network through cascading failure simulation experiments. The robustness is reflected by the ratio G of the number of subgraphs after the fault and the number of nodes of the largest connected subgraph in the network before and after the fault. n represents the number of network nodes before the fault, and nmax represents the maximum number of child graph nodes after the fault. The formula is expressed as:

\[ G = \frac{n_{\text{max}}}{n} \]

The importance ranking of the nodes is obtained according to various evaluation methods. The important nodes in the network are sequentially removed, and the two nodes S and G are compared. The smaller G is, the larger S is, indicating that the evaluation method is more reliable. Below we select the first five nodes from the evaluation results of the literature to delete the S and G values, and compare them with the S and G values obtained by deleting the first five nodes in this paper. The comparison results are shown in the following table.

Table 2 Comparative analysis of robust results

| | Reference 1 | Reference 2 | Reference 3 | Reference 4 | Our method |
|---|---|---|---|---|---|
| S | 6 | 6 | 3 | 5 | 6 |
| G | 10 | 10 | 14 | 7 | 7 |

It can be seen from the above table that after deleting the five most important nodes according to different ranking methods, the S value obtained by the evaluation method used in this paper is large, and the G value is significantly smaller than the G value obtained by other methods. The experimental results
show that the method of selecting key nodes in the network based on the construction tree proposed in this paper is reliable for evaluating the importance of nodes in the network.

5. CONCLUSION
This paper organizes the network architecture by evaluating the importance of the nodes in the network. Based on the directed weighted network model, the method of constructing the KD tree and the KB tree based on the graph structure is introduced in detail, and the depth and breadth impact is evaluated based on the obtained tree. This method not only considers the influence of the root node on other nodes in the construction tree, but also considers the influence of ignored nodes, and the evaluation is more accurate. The robust experimental analysis of the ARPA network proves the effectiveness of the method.

**Fund Project:**
- National Natural Science Foundation of China (71661015);
- Science and Technology Research Project of Jiangxi Provincial Department of Education (GJJ160330);
- Key Project of Soft Science Research in Jiangxi Province (20161BBA10015);
- Pilot project of comprehensive reform of general undergraduate higher education institutions in Jiangxi Province in 2014;
- 2014 Excellent Undergraduate University Excellent Engineer Education and Training Program in Jiangxi Province.

**REFERENCES**

[1] Hu P, Fan W L, Mei S W. Physica A: Stat. Mech. Appl. 2015
[2] Fan W L, Hu P, Liu Z G. IET Gener. Transm. Distrib.  2016
[3] Zhou X, Zhang FM, Li KW, Hui XB, Wu HS. Determination of key nodes of complex networks using importance evaluation matrix [J/OL]. Acta Phys. Sin., 2012, 61(05): 1-7.
[4] ZHAO YZ, WANG ZL, ZHENG J, GUO XJ. Determination of the most important nodes in communication networks by using importance contribution matrix [J]. Journal of Beijing University of Aeronautics and Astronautics, 2009, 35(09): 1076-1079.
[5] Newman M E J 2003 SIAM Rev. 45 167
[6] Batool K, Niazi M A. PLoSOne. 2014
[7] Barabasi A L, Bonabeau E. Scientific American. 2003
[8] Xu J, Li J X, Xu S. J. Zhejiang Univ.: Sci. C. 2012
[9] WU YY, XU G. Density characteristics and key node identification of heterogeneous dependent networks [J]. Acta Automatica Sinica, 2018, 44(05): 953-960.
[10] REN ZM, SHAO F, LIU JG, GUO Q, WANG BH. Study on the importance measurement method of network nodes based on degree and agglomeration coefficient [J]. Acta Physica Sinica, 2013, 62(12): 522-526.
[11] Kitsak M, Gallos L K, Havlin S, Liljeros F, Muchnik L, Stanley H E, Makse H A 2010 Nature Physics 11 22
[12] Wang Y, Guo J L. Method for importance evaluation of directional weighted network nodes based on multiple influence matrix [J/OL]. Acta Phys. Sin., 2017, 66(05): 19-30.
[13] ZHAO J, LIU W, LI X, WANG MD, MO SK. Evaluation method of node influence based on third degree theory [J]. Journal of Fuyang Teachers College, 2016, 33(04): 78-82.
[14] Codd E F 1970 Commun. ACM 13 377
[15] Yi Y R, Lao S Y, Wang Y D, Bai L, Chen L D. Identity evaluation algorithm for complex network nodes based on domain similarity [J]. Acta Physica Sinica, 2017, 66(03): 371-379.