# Chapter "Network Centralities"
# Solutions to exercises

Dirk Koschützki

Exercises:

1. Please *name* the centralities that can be computed for the four networks in Fig. 1.
2. Please *compute* the possible centralities for the four networks in Fig. 1. Choose sensible values for required parameters.
3. Explain why the centralities eccentricity and closeness cannot be computed for some of the networks in Fig. 1. Is it possible to compute shortest path betweenness for these networks?
4. Modify both algorithms for eccentricity such that closeness centrality is computed.
5. Compute the centralities Katz status index, eigenvector centrality and PageRank with CentiBiN or Visone for the networks in Fig. 1.
6. Search the literature for at least three more definitions of centralities.

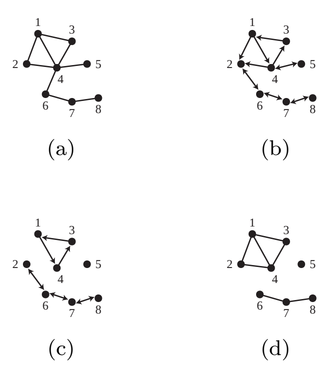


Figure 1: Four small networks for the exercises.

Solutions:

1. The networks are:

   (a) Undirected and connected

   (b) Directed and weakly connected

   (c) Directed and not connected

   (d) Undirected and not connect

   Therefore the following centralities described in this chapter can be applied:

   (a) Degree, Eccentricity, Closeness, SP-Betweenness, Katz Status Index, Eigenvector, and PageRank.

   (b) Degree, SP-Betweenness, Katz Status Index, and PageRank.

   (c) Degree, SP-Betweenness, Katz Status Index, and PageRank.

   (d) Degree, SP-Betweenness, Katz Status Index, and PageRank.

2. **Centralities for network (a)**

| | deg | ecc | clo | spb | katz | ev | pr |
|---|---|---|---|---|---|---|---|
| v1 | 3 | 0.25 | 0.071 | 0.5 | 150.4 | 0.481 | 0.150 |
| v2 | 2 | 0.25 | 0.066 | 0 | 119.2 | 0.381 | 0.104 |
| v3 | 2 | 0.25 | 0.066 | 0 | 119.2 | 0.381 | 0.104 |
| v4 | 5 | 0.33 | 0.1 | 15.5 | 188.3 | 0.601 | 0.252 |
| v5 | 1 | 0.25 | 0.065 | 0 | 66.2 | 0.212 | 0.062 |
| v6 | 2 | 0.5 | 0.083 | 10 | 77.0 | 0.247 | 0.118 |
| v7 | 2 | 0.33 | 0.063 | 6 | 32.0 | 0.099 | 0.133 |
| v8 | 1 | 0.25 | 0.045 | 0 | 11.5 | 0.035 | 0.075 |

**Centralities for network (b)**

| | in-deg | out-deg | spb | katz | pr |
|---|---|---|---|---|---|
| v1 | 1 | 2 | 6 | 0.617 | 0.053 |
| v2 | 3 | 1 | 12 | 2.447 | 0.176 |
| v3 | 1 | 1 | 2 | 0.764 | 0.040 |
| v4 | 2 | 3 | 9 | 1.184 | 0.075 |
| v5 | 1 | 1 | 0 | 0.764 | 0.040 |
| v6 | 2 | 2 | 12 | 2.190 | 0.267 |
| v7 | 2 | 2 | 8 | 1.811 | 0.232 |
| v8 | 1 | 1 | 0 | 0.985 | 0.117 |

**Centralities for network (c)**

| | in-deg | out-deg | spb | katz | pr |
|---|---|---|---|---|---|
| v1 | 1 | 1 | 1 | 0.538 | 0.140 |
| v2 | 1 | 1 | 0 | 0.896 | 0.098 |
| v3 | 1 | 1 | 1 | 0.538 | 0.140 |
| v4 | 1 | 1 | 1 | 0.538 | 0.140 |
| v5 | 0 | 0 | 0 | 0 | 0.021 |
| v6 | 2 | 2 | 4 | 1.559 | 0.182 |
| v7 | 2 | 2 | 4 | 1.559 | 0.182 |
| v8 | 1 | 1 | 0 | 0.896 | 0.098 |

**Centralities for network (d)**

| | deg | spb | katz | pr |
|---|---|---|---|---|
| v1 | 3 | 0.5 | 9.625 | 0.165 |
| v2 | 2 | 0 | 7.438 | 0.115 |
| v3 | 2 | 0 | 7.438 | 0.115 |
| v4 | 3 | 0.5 | 9.625 | 0.165 |
| v5 | 0 | 0 | 0 | 0.021 |
| v6 | 1 | 0 | 0.788 | 0.108 |
| v7 | 2 | 1 | 1.252 | 0.204 |
| v8 | 1 | 0 | 0.788 | 0.108 |

3. Eccentricity and Closeness can be computed for network (a) only as this is the only network that is strongly connected. Shortest-path betweenness does not require the network to be connected. Therefore it is applicable to all four networks.

4. First algorithm for closeness

**naive_eccentricity_centrality algorithm** (network $G = (V, E)$)

for each vertex $v \in V$

  $\mathcal{C}_{clo}(v) := 0$;

dist := compute_distance_matrix ($G$);

for each vertex $s \in V$

  for each vertex $t \in V$

    $\mathcal{C}_{clo}(s) := \mathcal{C}_{clo}(s) + \text{dist}(s, t)$;

  $\mathcal{C}_{clo}(s) := \frac{1}{\mathcal{C}_{clo}(s)}$;

return ($\mathcal{C}_{clo}$);

Second algorithm for closeness

```
bfs__eccentricity__centrality algorithm (network G = (V, E))
    for each vertex s ∈ V
        C_clo(s) := single__vertex__bfs__eccentricity (s, G);

single__vertex__bfs__eccentricity algorithm (vertex s, network G = (V, E))
    result := 0;
    for each vertex v ∈ V
        distance (v) := -1;
    distance (s) := 0;
    list L = [s];
    while L is nonempty {
        remove vertex w from the front of list L;
        for each neighbor x of w {
            if (distance (x) == -1) {
                distance (x) := distance (w) + 1;
                result := result + distance (x);
                add x to the end of list L
            }
        }
    }
    return 1/result;
```

5. For results see answer to exercises 2.

6. The CentiBiN-Paper has a huge list of references. For example:

   D. Koschützki, H. Schwöbbermeyer, and F. Schreiber: Ranking of network elements based on functional substructures. Journal of Theoretical Biology 2007, Volume 248, Issue 3, Pages 471–479.

   U. Brandes and D. Fleischer: Centrality Measures Based on Current Flow. Proceedings 22nd Symposium Theoretical Aspects of Computer Science (STACS 2005), Volume 3404 of Lecture Notes in Computer Science (LNCS), Springer 2005, Pages 533–544.

   J. M. Kleinberg: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM 1999, Volume 46, Issue 5, Pages 604–632.