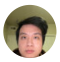
Recursion algorithm with node centrality in social networks

Preprint · January 2024
DOI: 10.13140/RG.2.2.20855.19367



Zhou Zhou
New York University

Recursion algorithm with node centrality in social networks

Zhou Zhou[a,1]

[a]Department of Mathematics, University at Buffalo, Buffalo, NY 14261

The real-world social network undertakes an array of complex functions that shape different people's relationships and built diverse relations between individuals. Friendship is very important and common in society. The social network also indicates the small-world effect. Stanley Milgram's experiment[J.Travers, S.Milgram, Sociometry 32, 425–443 (1969)] (1) uses mail to learn more about the probability that two randomly selected people would know each other and attempt to find the average path length between two persons. In general, Breadth-first search (BFS) can find the shortest path between two nodes and this algorithm is an exhaustive method. My algorithm only allows each user to send once to an adjacent user until it reaches the target node or failed. For example, if I want to send a mail to UB President Satish K. Tripathi, which friend I should choose to ensure UB President can receive my mail as soon as possible. To achieve this goal, I hope this letter passed as few times as possible. I use the Facebook dataset (ego-Facebook), which includes every user's friend list, to analyze the relations between node centralities and the path between two nodes. I construct a recursion algorithm to find the path between two nodes following their node centrality. My result indicates that the eigenvector centrality is the best method to deliver mail in my experiment. I also propose 3 types of people with different characteristics for analysis and choose the most suitable centrality to pass mail.

complex networks | small world effect | node centrality | social network

Social networks have undergone an exponentially rapid with development of social media application (2–8). Social network in real life now arguably encompasses many areas from psychological and cognitive sciences to online world (9, 10). Consequently, social networks are now commonly regarded as an essential substrate for studying many other facets in network science (11–16), such as social scientists to model social networks and study the behavior (17, 18) and social contagion (19).

Facebook remains the most popular social media in 2016 (20). Many researchers use Facebook's anonymous dataset and data to construct model and attempt build relations with real-world (21–29). For example, they analyze gender inequality through large-scale Facebook advertising data (30). I analyze Stanford Large Network Dataset. It is a social network and its name is ego-Facebook (31). This dataset consists of 'circles' (or 'friends lists') from Facebook users. Facebook data was collected from survey participants using this Facebook app (32). I use Python to analyze the network data, specifically the networkx module to construct the network and I also will use the matplotlib module to draw the image. This social network has total 4,039 nodes and 88,234 edges. Figure 1 indicates this Facebook social network dataset network image. It is an undirected and unweighted social network. The average clustering coefficient is 0.6055, the average shortest path is 3.6925. Watts and Strogatz model (33) indicates the social network is a kind of small-world network. In this case, social network neighbors of any given node are likely to be neighbors of each other and most nodes can be reached from every other node by a small number of steps.

In my experiment, I try to send a mail to UB President Satish K. Tripathi and he can receive my mail as soon as possible. And every participant only can deliver mail once to another person until to UB President. I use degree centrality, closeness centrality (34), betweenness centrality (35), eigenvector centrality and PageRank centrality as node weight to find the shortest path between two nodes. The degree centrality measure for a node in a network is just its degree, the number of edges connected to it (36). Closeness centrality is a centrality score that measures the mean distance from a node to other nodes (36). Betweenness centrality measures the extent to which a node lies on paths between other nodes (36). Eigenvector centrality awards a number of points proportional to the centrality scores of the neighbors (36).

BFS (Breadth-first search) starts at the initial node and explores all nodes at the present depth prior to moving on to the nodes at the next depth level. In fact, this algorithm uses brute force to traverse nodes and edges to find the shortest path between two nodes. The runtime of BFS is O(N+E), where N stands for vertices and E stands for edges. In the practice, it will cost much time and effort to send mails to all friends. In this case, I designed a recursion algorithm to simulate the delivery process in the real world. I choose the adjacent node as every recursion initial node, which has the maximum node centrality in this node list. This algorithm will finally find the target node or failed. The runtime of my

Significance Statement

Social network is a common network in the real world and individuals in the social network are connected by friendship. It is hard for people to make friends without any commonalities. In my experiment, eigenvector centrality can be considered an effective way to socialize with other people. In this case, understanding node centrality is not only for making friends but also very helpful to an individual's self-development. Besides, my result also indicates that closeness centrality also can achieve the shortest path between two nodes. In practice, the complexity of eigenvector centrality is much lower than other node centrality. Individuals just need to send mail to their friend who has some famous friends and loop this operation to deliver mail to the target person.

Author contributions: Z.Z. designed research, performed research, analyzed data and wrote the paper.

The author declares no competing interest.

[1]To whom correspondence should be addressed. E-mail: zzhou38@buffalo.edu

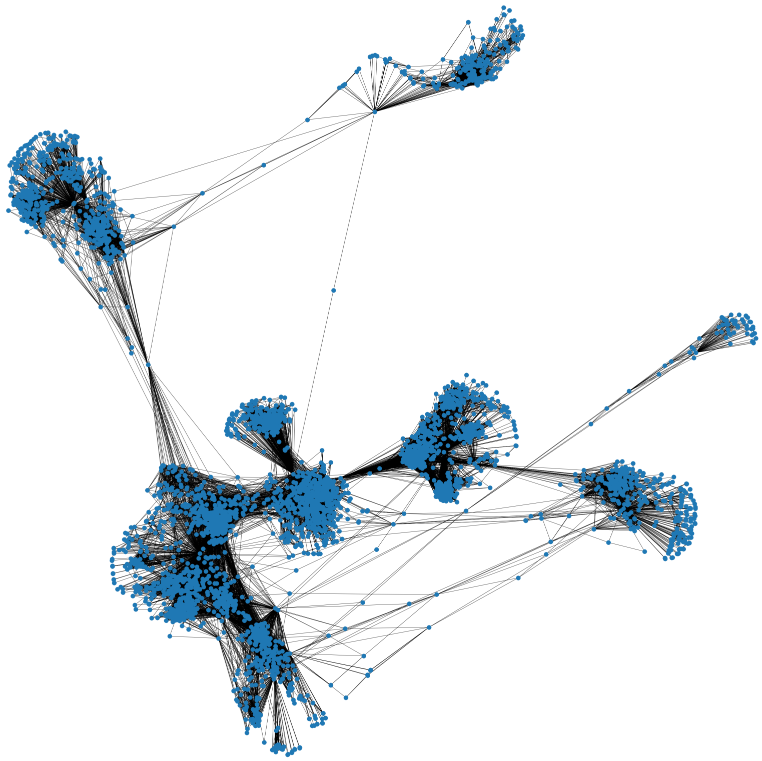


Fig. 1. This figure is the visualization of network. Each blue nodes represent a Facebook user and each black edges represent user's Facebook friend list. This figure includes 4,039 nodes and 88,234 edges.

recursion algorithm is O(N).

I find that closeness centrality is the best to find the shortest path. However, it is also the least realistic method to deliver mail. The effective order: eigenvector centrality > betweenness centrality > closeness centrality > degree centrality > PageRank centrality. In the practice experiment, the degree centrality and eigenvector centrality may be the most convenient way to deliver mail between participants.

Result. I use the Facebook dataset to construct the social network. Then I use the recursion algorithm 2, which lets the largest centrality adjacent node as the next recursion initial node and find the path between two nodes. Besides, I use the Breadth-first search (BFS) algorithm to find the shortest path between two nodes and compare my recursion algorithm result. The shortest path between two nodes becomes a gold standard and references.

First, I find the largest and minimal centrality nodes through different methods (Table 1). These nodes represent some special individuals in the real world. For example, the largest degree centrality node (Node-id: 107) represents that the person has the maximum number of friends in this social network, otherwise the minimal degree centrality node (Node-id: 11) has the fewest friends. Furthermore, Node-id 107 also is the largest closeness centrality node and it is closer to all nodes. The betweenness centrality also provides 107 as the largest node and this means that node 107 is in the other nodes' shortest path. This node becomes an important bridge to connect with other nodes. The largest eigenvector node (Node-id: 1912) represents that the individual has many famous friends although the total number of his friends may be less than some persons. Besides, Node-ID 3437 has the largest PageRank centrality so many users connect to him. These 3 nodes represent 3 special persons and I will analyze these largest centrality nodes to explore the best method to send mail.

Then, I use my recursion algorithm 2 to analyze these special nodes based on node centrality. In a complete network, BFS algorithm can find the shortest path between two nodes. My algorithm only uses the largest centrality adjacent node as the next recursion initial node, in this case, some nodes cannot successfully reach the target node. In this case, I add "END" as a sign of the end path between two nodes. I also filter a special situation, which represents the first initial node and the target node adjacent. It means the first initial node

Table 1. Largest and Minimal centrality nodes in different methods

Node centrality	Largest node ID	Minimal node ID
1. Closeness centrality	107	692
2. Betweenness centrality	107	11
3. Eigenvector centrality	1912	692
4. PageRank centrality	3437	2079
5. Degree centrality	107	11

It may exist multiple nodes in the Largest node and Minimal node. In this table, I just indicate one node. Due to the Facebook anonymous dataset, I use Node-id to represent every individual.

Table 2. Node Centrality recursion path V.S BFS shortest path with target Node-Id "107"

Node Centrality	Total nodes	Larger nodes	BAP/RAP
1.Closeness centrality	2993	0	3.58/3.58
2.Betweenness centrality	2993	254	3.58/3.68
3.Eigenvector centrality	2222	713	3.53/3.85
4.PageRank centrality	2387	956	3.47/12.49
5.Degree centrality	2935	1504	3.55/27.39

Total nodes filter all path length equal 2 nodes and some nodes which cannot reach the Node-id 107. The BAP represents BFS average path and the RAP represents the recursion algorithm average path

Table 3. Node Centrality recursion path V.S BFS shortest path with target Node-Id "1912"

Node Centrality	Total nodes	Larger nodes	BAP/RAP
1.Closeness centrality	3283	0	4.27/4.27
2.Betweenness centrality	2505	2375	4.36/5.94
3.Eigenvector centrality	3283	17	4.2747/4.2799
4.PageRank centrality	2676	2676	4.11/146.84
5.Degree centrality	3225	3225	4.26/181.35

Total nodes filter all path length equal 2 nodes and some nodes which cannot reach the Node-id 1912.

is already connected to the target node and it does not need any necessary node centrality to guide the pathway.

In my experiment, I choose Node-ID 107 as my target node followed by different node centrality in the recursion algorithm and other nodes become the initial nodes. I use all available nodes to reach the target node through the recursion algorithm. In daily life, Node-ID 107 represents the user who has the maximum number of friends. Table 2 indicates the results. It shows the degree centrality average length between one node and the largest centrality node is 27. Although the eigenvector centrality still has 713 nodes whose path is larger than BFS algorithm path, the average path is 3.85, which is very close to the BFS average shortest path. The closeness centrality is based on the shortest path to reward every node so that my recursion algorithm can be considered as a BFS algorithm to find the pathway. In this case, the larger node of closeness centrality in table 2 is 0 and BAP/RAP is the same number (3.58/3.58).

Table 3 indicates the largest eigenvector node "1912" as the target node result. The Node-ID 1912 means the user has some famous friends and he may not have many friends like Node-ID 107. Then the PageRank centrality and degree centrality have a very large average length than other centrality. All valid nodes' pathways are larger than BFS nodes. It means that these two methods will spend more time than other centrality methods. In contrast, betweenness centrality also has 2375 larger nodes than BFS nodes but it keeps low recursion algorithm average path. The closeness centrality still keeps 0 larger nodes and the same BAP/RAP (4.27/4.27). The eigenvector centrality average path length is also close to the BFS average path length.

Finally, table 4 illustrates the Node-ID 3437 as the target node result, which is the largest PageRank node. The closeness centrality and betweenness centrality cannot keep low average lengths like table 2 and table 3. Eigenvector centrality and BFS still keep a low average path than other centralities but the

Table 4. Node Centrality recursion path V.S BFS shortest path with target Node-Id "3437"

Node Centrality	Total nodes	Larger nodes	BAP/RAP
1.Closeness centrality	3491	3285	4.52/15.62
2.Betweenness centrality	3491	2464	4.5/7.23
3.Eigenvector centrality	206	206	3.68/3.68
4.PageRank centrality	1733	1554	4.84/166.40
5.Degree centrality	208	173	3.69/15.50

Total nodes filter all path length equal 2 nodes and some nodes which cannot reach the Node-id 3437.

total nodes of eigenvector centrality and degree centrality only have 206/208 nodes. It indicates that there are almost 95% nodes that cannot successfully reach the Node-ID 3437 through these two centrality methods. In other words, the letters may be lost in the delivery process. PageRank centrality recursion algorithm average length is 166.40. Then I sort ascending these centralities based on their recursion algorithm average path results, eigenvector centrality $<$ betweenness centrality $<$ closeness centrality $<$ degree centrality $<$ PageRank centrality. The eigenvector centrality and degree centrality may lose mail for some special targets.

Discussion. The help of friends contributes to personal development. Currently, social media also help people to make friends and it can be considered as 'add edges' in a social network. I use the Facebook dataset to construct a small social network and a recursion algorithm to find a path between two nodes. I find 3 types of people based on my result (table 1). First, this type of person has the most friends in the social network. Second, this type of person has some very famous friends in this social network. Finally, this type of person has many fans like a movie star.

Table 2, table 3 and table 4 indicate that eigenvector centrality is the best method in my experiment. In other words, if I want to deliver mail to UB president, I will choose a person who is my most influential friend or who has several influential friends in daily life. Then my friends also follow my strategy to pass mail to their friends. If they found someone who is already on the pathway, they just need to pass mail to the second famous friend in their friend list. In this case, the UB president will receive my mail as soon as possible. In this process, every participant only needs to pass mail once to the next participant. However, if someone wants to choose Taylor Swift as the target person based on eigenvector centrality, my result also indicates that this letter almost cannot reach her (95% loss mail). This guy should follow betweenness centrality to deliver mail. In practice, he should choose his friend who is good at socializing. Then every participant follows this strategy to pass mail. Finally, if I want to deliver mail to someone who has many friends in daily life, I still follow eigenvector centrality because it has a shorter average pathway than degree centrality.

I design another version of Milgram's experiment and I use the recursion algorithm to simulate it in the social network. People should follow different node centrality based on diverse types of people. I give up closeness centrality because people cannot know the shortest path between two persons in the real world. It is just an ideal method and it has no practical significance.

Materials and Methods

Data. My data came from the online Facebook application. I use this data as my primary motivation to analyze the social network. The network includes 4,039 nodes and 88,234 edges, which represent each Facebook user's friend list. Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value (37). Also, while feature vectors from this dataset have been provided, the interpretation of those features has been obscured (37). For instance, where the original dataset may have contained a feature "political=Democratic Party", the new data would simply contain "political=anonymized feature 1" (37). Thus, using the anonymized data it is possible to determine whether two users have the same political affiliations, but not what their individual political affiliations represent (37).

Closeness Centrality. In my experiment, I use different node centrality to determine node weight. The equation 1 is the closeness centrality.

$$C_i = \frac{1}{l_i} = \frac{n}{\sum_j d_{ij}} (36) \qquad [1]$$

The d_{ij} is the shortest distance from node i to node j. Closeness centrality calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. In this case, the more central a node is, the closer it is to all other nodes.

Betweenness Centrality. The equation 2 shows the betweenness centrality equation, which measures the extent to which a node lies on paths between other nodes.

$$C_i = \sum_{st} \frac{n^i_{st}}{g_{st}} (36) \qquad [2]$$

The n^i_{st} is the number of shortest paths from s node to t node that pass through i node. The g_{st} is the total number of shortest paths from s node to t node. In general, the betweenness centrality for each node is the number of these shortest paths that pass through the node.

Eigenvector Centrality. The equation 3 describes the eigenvector centrality. Newman describes the eigenvector centrality awards a number of points proportional to the centrality scores of the neighbors (36). Specifically, it assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. (38, 39). For the given graph $G(N, E)$ to construct an adjacency matrix $A(i, j)$. If node i is linked to node j the $a_{i,j} = 1$, otherwise the $a_{i,j} = 0$.

$$x_i = \frac{1}{k} \sum_{j \in M(i)} x_j = \frac{1}{k} \sum_{j \in G} a_{i,j} x_j \qquad [3]$$

The $M(i)$ is a set of the neighbors of i and k is a constant. Then the k can be found by eigenvector equation.

$$Ax = kx \qquad [4]$$

PageRank Centrality. The equation 5 indicates the PageRank centrality, which is a name given it by the Google web search corporation. Google uses PageRank as a central part of their web ranking technology for web searches, which estimates the importance of web pages and hence allows the search engine to list the most important pages first (40).

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j} \qquad [5]$$

The Google search engine uses a value of $\alpha = 0.85$ in its calculations (39). The k_j is the number of neighbors of node j. The difference between PageRank and eigenvector centrality is that the PageRank vector is a left-hand eigenvector. Besides, my Facebook dataset social network is an unweighted and undirected graph. In this case, I transfer the undirected graph to a directed graph with two directed edges for each undirected edge.

Degree Centrality. The equation 6 represents degree centrality. The simplest centrality measure for a node in a network is just its degree, it indicates the number of edges connected to the node.

$$x_i = deg(i) \qquad [6]$$

BFS(Breadth First Search) Algorithm. Breadth-first search (BFS) 1 is a method used for traversing a graph. It starts at any starting node in a graph and explores all of the neighbor nodes at the present depth before to moving on to the items at the next depth level. It provides the shortest path between two nodes, which is the “standard answer”. I can use this answer to compare my recursion result.

Algorithm 1 Breadth First Search Algorithm find the shortest path

Require: Unweighted directed graph $G = (V, E)$ as an adjacency list, vertices s, t
  Set $\texttt{dist}[u] = \infty$ for all u
  Set $\texttt{prev}[u] =$ NULL for all u
  Create queue Q
  Set $\text{dist}[s] = 0$
  Q.append(s)
  while Q.size() > 0 **do**
    Let $u = Q$.pop()
    for each out-neighbor v of u **do**
      if $\texttt{dist}[v] = \infty$ **then**
        Set $\texttt{dist}[v] = \texttt{dist}[u] + 1$
        Set $\texttt{prev}[v] = u$
        Q.append(v)
  Run and return Reconstruct along with $\texttt{dist}[t]$

Recursion Algorithm. A recursive algorithm calls itself with smaller input values and returns the result for the current input by carrying out basic operations on the returned value for the smaller input. The algorithm 2 indicates my recursion Algorithm.

Algorithm 2 Recursion algorithm to find path between two nodes based on node centrality

Require: $iNode, tNode, cPath, ncList$
Ensure: $adjnode = neighbors(iNode)$
  if $tNode \in adjnode$ **then**
    $cPath.append(tNode)$
    return $cPath$
  else
    $newlist = Map()$
    for $eachiteam \in adjnode$ **do**
      $newlist[eachiteam] = ncList[eachiteam]$
    $descendingSort(newlist.values())$
    for $eachkey \in newlist$ **do**
      if $eachkey \notin cPath$ **then**
        $cPath.append(eachkey)$
        $newNode = newlist[eachkey]$
        $BREAK$
    if $newNode ==$ "" **then**
      $cPath.append(\text{"}END\text{"})$
      return $cPath$
    return $recAlgorithm(newNode, tNode, cPath, ncList)$

The $iNode$ represents the current node and $tNode$ means the target node, which we want to find a path between $iNode$ to $tNode$. The $cPath$ means the current path list, which will record the current path we have already passed. Besides, it includes the initial node before recursion algorithm. The *ncList* is a map structure. It includes related node name and centrality.

1. SM Jeffrey Travers, An experimental study of the small world problem. *Sociometry* **32**, 425–443 (1969).
2. L Freeman, The development of social network analysis. *A Study Sociol. Sci.* **1**, 159–167 (2004).
3. J Scott, Social network analysis: developments, advances, and prospects. *Soc. network analysis mining* **1**, 21–26 (2011).
4. AP Velenturf, PD Jensen, Promoting industrial symbiosis: Using the concept of proximity to explore social network development. *J. Ind. Ecol.* **20**, 700–709 (2016).
5. O Serrat, Social network analysis in *Knowledge solutions.* (Springer), pp. 39–43 (2017).
6. J Scott, *What is social network analysis?* (Bloomsbury Academic), (2012).
7. S Leinhardt, *Social networks: A developing paradigm.* (Elsevier), (2013).
8. KA Fredericks, MM Durland, The historical evolution and basic concepts of social network analysis. *New directions for evaluation* **2005**, 15–23 (2005).
9. AD Kramer, JE Guillory, JT Hancock, Experimental evidence of massive-scale emotional contagion through social networks. *Proc. Natl. Acad. Sci.* **111**, 8788–8790 (2014).
10. M Szell, R Lambiotte, S Thurner, Multirelational organization of large-scale social networks in an online world. *Proc. Natl. Acad. Sci.* **107**, 13636–13641 (2010).
11. K Börner, S Sanyal, A Vespignani, , et al., Network science. *Annu. rev. inf. sci. technol.* **41**, 537–607 (2007).
12. U Brandes, G Robins, A McCranie, S Wasserman, What is network science? *Netw. science* **1**, 1–15 (2013).
13. T Tiropanis, W Hall, J Crowcroft, N Contractor, L Tassiulas, Network science, web science, and internet science. *Commun. ACM* **58**, 76–82 (2015).
14. S Havlin, et al., Challenges in network science: Applications to infrastructures, climate, social systems and economics. *The Eur. Phys. J. Special Top.* **214**, 273–293 (2012).
15. DA Scheufele, NM Krause, Science audiences, misinformation, and fake news. *Proc. Natl. Acad. Sci.* **116**, 7662–7669 (2019).
16. CA Bail, et al., Exposure to opposing views on social media can increase political polarization. *Proc. Natl. Acad. Sci.* **115**, 9216–9221 (2018).
17. TG Lewis, *Network science: Theory and applications.* (John Wiley & Sons), (2011).
18. JH Fowler, NA Christakis, Cooperative behavior cascades in human social networks. *Proc. Natl. Acad. Sci.* **107**, 5334–5338 (2010).
19. J Ugander, L Backstrom, C Marlow, J Kleinberg, Structural diversity in social contagion. *Proc. national academy sciences* **109**, 5962–5966 (2012).
20. S Greenwood, A Perrin, M Duggan, Social media update 2016. *Pew Res. Cent.* **11**, 1–18 (2016).
21. K Lewis, J Kaufman, M Gonzalez, A Wimmer, N Christakis, Tastes, ties, and time: A new social network dataset using facebook. com. *Soc. networks* **30**, 330–342 (2008).
22. SA Catanese, P De Meo, E Ferrara, G Fiumara, A Provetti, Crawling facebook for social network analysis purposes in *Proceedings of the international conference on web intelligence, mining and semantics.* pp. 1–8 (2011).
23. N Akhtar, H Javed, G Sengar, Analysis of facebook social network in *2013 5th International Conference and Computational Intelligence and Communication Networks.* (IEEE), pp. 451–454 (2013).
24. F Alam, EA Stepanov, G Riccardi, Personality traits recognition on social network-facebook in *Proceedings of the International AAAI Conference on Web and Social Media.* Vol. 7, pp. 6–9 (2013).
25. PW Fong, M Anwar, Z Zhao, A privacy preservation model for facebook-style social network systems in *European symposium on research in computer security.* (Springer), pp. 303–320 (2009).
26. K Lewis, M Gonzalez, J Kaufman, Social selection and peer influence in an online social network. *Proc. Natl. Acad. Sci.* **109**, 68–72 (2012).
27. TQ Phan, EM Airoldi, A natural experiment of social network formation and dynamics. *Proc. Natl. Acad. Sci.* **112**, 6595–6600 (2015).
28. W Youyou, M Kosinski, D Stillwell, Computer-based personality judgments are more accurate than those made by humans. *Proc. Natl. Acad. Sci.* **112**, 1036–1040 (2015).
29. CM Astley, et al., Global monitoring of the impact of the covid-19 pandemic through online surveys sampled from the facebook user base. *Proc. Natl. Acad. Sci.* **118**, e2111455118 (2021).
30. D Garcia, et al., Analyzing gender inequality through large-scale facebook advertising data. *Proc. Natl. Acad. Sci.* **115**, 6958–6963 (2018).
31. J Leskovec, A Krevl, SNAP Datasets: Stanford large network dataset collection (http://snap.stanford.edu/data) (2014).
32. J Leskovec, J Mcauley, Learning to discover social circles in ego networks in *Advances in Neural Information Processing Systems*, eds. F Pereira, C Burges, L Bottou, K Weinberger. (Curran Associates, Inc.), Vol. 25, (2012).
33. DJ Watts, SH Strogatz, Collective dynamics of 'small-world' networks. *Nature* **393**, 440–442 (1998).
34. A Bavelas, Communication patterns in task-oriented groups. *The journal acoustical society Am.* **22**, 725–730 (1950).
35. LC Freeman, A set of measures of centrality based on betweenness. *Sociometry* pp. 35–41 (1977).
36. MEJ Newman, *Networks: an introduction.* (Oxford University Press, Oxford; New York), (2010).
37. J Leskovec, J Mcauley, Learning to discover social circles in ego networks. *Adv. neural information processing systems* **25** (2012).
38. CF Negre, et al., Eigenvector centrality for characterization of protein allosteric pathways. *Proc. Natl. Acad. Sci.* **115**, E12201–E12208 (2018).
39. MEJ Newman, *Mathematics of Networks.* (Palgrave Macmillan UK, London), pp. 1–8 (2016).
40. S Brin, L Page, The anatomy of a large-scale hypertextual web search engine. *Comput. networks ISDN systems* **30**, 107–117 (1998).