The Turtles Relationships

Big Graphs Analysis

Sofia Begonha Morgado

Faculdade de Ciências e Tecnologia

Universidade Nova de Lisboa

June 3, 2022

Contents

1 Introduction

Social animals cohabit and interact, establishing intricate social structures and relationships. Typically, studies do not clearly address these relationships, but instead focus on other also relevant aspects. Social network analysis is the study of social structures and relationships as networks of nodes connected by edges, corresponding to social ties. The use of social network analysis to study animal behavior can improve the biological field by finding and quantifying certain features of social relationships, many of which are not represented by more prevalent measures of sociality, such as group size [1].

In order to better understand tortoise interactions, the Reptilia Tortoise Network [2] was selected. This graph consists of an animal social network where nodes correspond to Nevada desert tortoises from the species *Gopherus agassizii* and the edges are relationships between individuals.

The graph refers to a temporal network in which edge timestamps correspond to the interaction year. The data was collected within 7 years. However, for the most of this investigation, the timestamp will not be considered.

Important characteristics of this network, including the degrees of the nodes and density of the network; paths and walks, as well as the diameter and eccentricity; the components of the graph; the centrality, cliques, and cores; and clustering characteristics, will be addressed with the aid of appropriate methodologies.

2 Methods

To undertake a comprehensive graph analysis, a network was chosen from the website networkrepository.com. The reptilia-tortoise-network-pv graph was chosen from the available Animal Social Networks. Firstly, the data concerning the year of interaction was removed, and a graph containing multiple edges between some pairs of nodes was used in this project.

Using R's Igraph package's functionalities, an extensive graph analysis was implemented.

3 Results and Discussion

The Tortoise Relationship Graph contains 35 vertices, corresponding to 35 turtles, and 104 edges, which correspond to the relationships. As we may see, this is an undirected and unweighted network. A representation of the graph is shown in Figure 1.

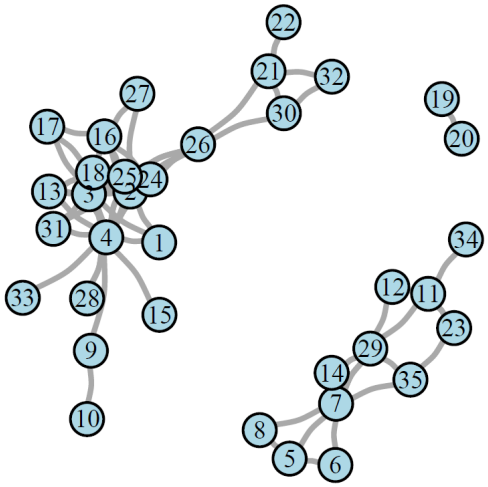

Figure 1: Graph with turtle relationships

Several functions from Igraph R package that may be used to show the edges. For example, as_edgelist() which produces a simple list of the edges, or as_adj_list() and as_adj_edge_list() which show, for each node, the list of vertices connected ans the list of edges connected to it, respectively.

3.1 Adjacency Matrix

We may represent this network using an adjacency matrix [3]. The adjacency matrix is the fundamental mathematical representation of a network, defined to be a $n * n$ matrix showing connections between nodes, so that for nodes i and j, $A_{ij} = e$, where e is the number of edges connecting them. As this is a graph without self-edges, the diagonal matrix elements are all 0. Also, we may see that there are multi-edges, meaning, nodes connected with more than one edge. This may be due to the fact that this network shows interactions between turtles in different years, and the same two tortoises may interact in different years. In fact, we could display this data in several subgraphs for each year, in order to constitute a dynamic or temporal network [4]. Nevertheless, in this project this network will be addressed as a single graph. Concluding, we classify this graph as an multi-graph [4]. Also, with this information we may conclude that the same turtles may have interactions throughout the years.

3.2 Nodes degrees

The degree of a node is the number of edges connected to it and is one of the most useful network concepts [4]. In this case, it corresponds to the number of interactions each tortoise has had (and not

to the number of tortoises each tortoise as interacted with, as the degree correspond to the number of edges and not the number of neighboring nodes). We may access data concerning the degree by the sum of each row of the adjacency matrix, or with built-in Igraph functions.

The maximum degree of this graph is 20 and the mean degree is 5.94. Also, the minimum degree is 1, which means some tortoise only contacted with another tortoise in a single year, during the 7 year study.

An histogram of the degree distribution is presented in Figure 2.

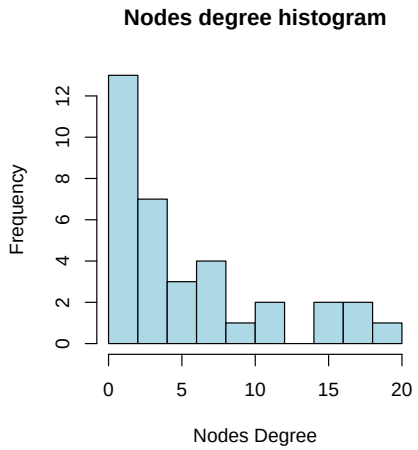

Figure 2: Histogram of vertices degree

Occasionally, the degree distribution's tail diminishes as a power of k. If the degree distribution follows a power law, then the cumulative distribution function, which is the proportion of nodes with degree k or more, likewise follows a power law. For this graph, when fitting the power law, we obtain an exponent of 8.356 and a p-value of the Kolmogorov-Smirnov test of 1, meaning we cannot conclude there is a power law in this case. This is due to the fact that this graph has a small number of vertices. The distribution and cumulative distribution of the nodes degree is shown in Figure 3.

3.3 Density and Sparsity

The density of a network is the fraction of existing edges from the maximum possible number of edges, and varies between 0 and 1 [4].

Since this is a multi-graph containing multi-edges corresponding to interactions in each year, the density may be calculated with two methodologies. First, the density for the corresponding simple graph, produced by using the simplify() function, was calculated and takes a value of 0.111. On the other hand, the mean density obtained for each year, calculated as the total density divided by the number of years, corresponds to 0.025.

A network is considered dense if, as the number of edges becomes larger, its density remains non-zero. Since it is not possible to estimate this for real world networks, there is no formal sense to classify

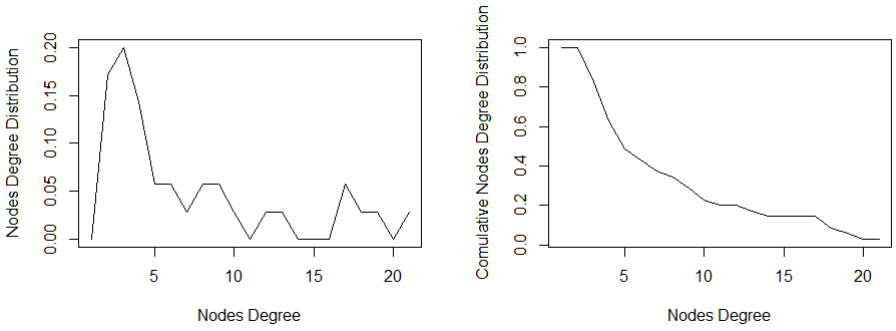

Figure 3: Distribution of the degree of the nodes and Right-Tail Cumulative distribution of the degree of the nodes

a network as dense or sparse [4]. Nevertheless, we may observe low values of density in this graph, especially when considering the mean density for each year.

With this information we may speculate that this group of tortoise is not a unique community, or that tortoise do not live in communities at all, questions which should be addressed to a specialist in the matter.

3.4 Paths and Walks

A walk in a network is any series of nodes in which each pair of successive nodes is linked by an edge.

The shortest path is one that traverses the fewest edges [4].

As this is an unconnected graph, we must only calculate the paths between nodes in the same component. For example, we may study one of the shortest paths between nodes 10 and 32 using the shortest_path() function. This will retrieve the ordered set of nodes and the list of edges of this path. In this case, the result obtained was: 10, 9, 4, 2, 26, 21, 32. We may plot the results, as is seen in Figure 4.

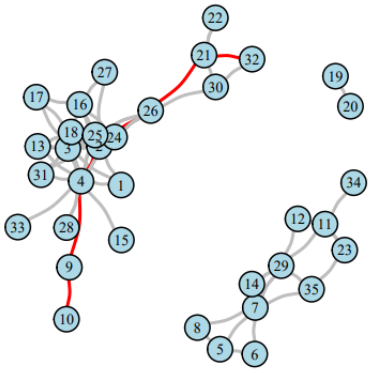

Figure 4: Shortest Path from node 10 to node 32

Studying the paths and walks may be useful for example for determining the probability of diseases'

spreading, even between two tortoises that have no direct contact.

3.5 Diameter and Eccentricity

The diameter of a graph corresponds to the longest shortest path [4] and may be calculated using the diameter() function. In graphs with many components, the diameter will be the longest shortest path existing in all of the components. The diameter of this graph is 6.

Also, we may calculate, for each node, the shortest path distance from the farthest other node in the graph using the function eccentricity(). In this graph, eccentricity varies from 1 to 6.

The mean distance cannot be calculated in an unconnected graph as it is not possible to determine the distance between two vertices from different components.

3.6 Components

Networks may have two or more separate parts that are disconnect from one another, called components, defined so that there exists a path from each component member to every other component member [4]. We may use the function is_connected() to evaluate the presence of separated components in a network, and the function components() to characterize that components. As we may see, this graph contains 3 components, with 22, 11 and 2 nodes, respectively. This means that the fraction of nodes in the main component of this graph corresponds to 0.628.

Using the function decompose() we may separate the graph into its three components. The 3 separate components are shown Figure 5 by the usage of different shades of blue.

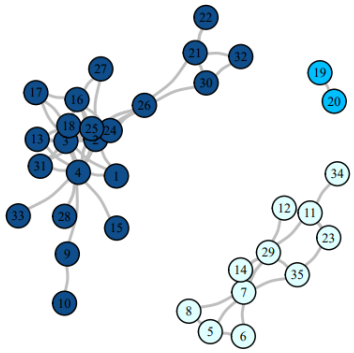

Figure 5: Graph with tortoise relationships comprising 3 separate components

The presence of this components may mean that the tortoises interact in groups or families. Possibly, there are even other tortoises interacting with the ones in the smaller components, and that are missing in the data set. Also, it would be useful to understand if the component composed of only two tortoises comprises a male and a female turtle.

3.7 Connectivity and Cut sets

Connectivity is the amount of independent pathways between two nodes and is a measure of how strongly they are linked [4]. For example, when vertices 1 and 25 are chosen, their connectivity is equal to 3, when selecting vertices 1 and 15, their connectivity is 1. This means tortoise number 1 and 25 are more linked than 1 and 15.

A cut set is a collection of nodes and neighboring edges whose removal will result in the disconnection of two nodes [4]. For our previous examples of connectivity, we may observe that the cut set is equivalent to the connectivity.

3.8 Centrality

A simple but useful measure of a node's centrality in a network is its degree, or the number of edges connecting to it, and is called degree centrality [4]. The degree centrality is shown in Figure 6.

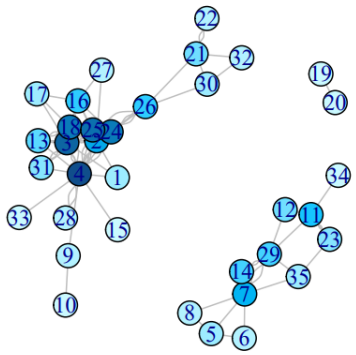

Figure 6: Centrality based on node degree. Nodes with higher degrees are in dark blue, whilst those with lower degrees are represented in light blue

Nevertheless, degree is a somewhat rudimentary indicator of centrality. Other measurements may be applied. For example, the eigenvector centrality, which takes into account not only how many edges a node has, but also the importance of the nodes it is linked to, based on their centrality [4]. The eigenvector centrality is shown in Figure 7. As we may see, some nodes previously with intermediate levels of centrality, as for example, 7, 11, 14 and 29, are now given a lower score of centrality as they are connected to nodes with low level of centrality.

Finally, a distinct notion of node significance is conveyed by betweenness centrality, which quantifies the amount to which a node is located on pathways connecting other nodes. It may be calculated with the function centr_betw(). Despite the fact that it is conceivable, it is uncommon to utilize betweenness centrality to compare nodes in distinct components, as is the case in this graph [4].

There are more centrality metrics available, but they must be applied to graphs that respect their application constraints. For example, the closeness centrality, a centrality number that reflects the average distance between nodes, must not be used to disconnected networks. Also, the Kartz, PageRank

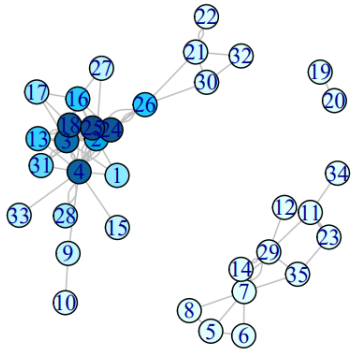

Figure 7: Eigenvector centrality. Nodes with higher centrality score are in dark blue, whilst those with lower score are represented in light blue

and Hubs and Authorities, which are usefull centrality measurements for directed graphs [4].

3.9 Cliques and Cores

Cliques are sets of nodes in an undirected graph so that all the members are connected with an edge and it is usually an indication of a highly cohesive subgroup [4].

A maximal clique is one that cannot be extended, and may even include every pair of two connected nodes. They can be shown with the usage of the max_cliques() function. Conversely, the largest cliques in a network are those that include the greatest number of nodes,and are always maximal.

Figure 8 displays a graph highlighting the two greatest cliques. Both consist of six nodes, and they share five of those nodes. This may indicate a colony of tortoises that interact with one another.

With this information we may also conclude that this graph is not planar, because it includes at least one subgraph homeomorphic to a complete (full) graph with 5 vertices (K5).

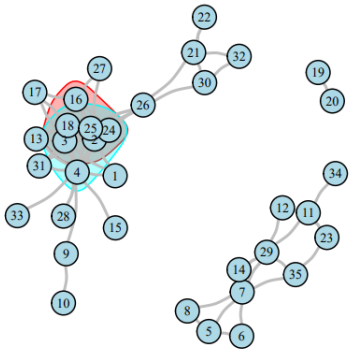

Figure 8: Graph enhancing the two largest cliques in the graph

A k_core is largest connected subgraph in which every vertex has at least degree k. Using the coreness() method, we determine the coreness of each vertex. Highlighted in Figure 9 are the cores with

$k \geq 7$. This is the same location shown in Figure 8, which depicts this integrated tortoise community.

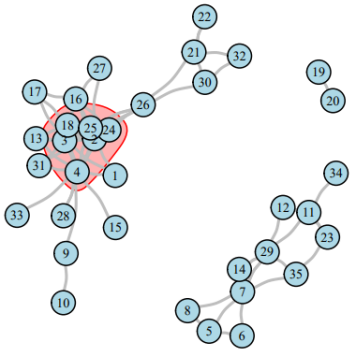

Figure 9: Graph enhancing cores with k$\geq$ 7

3.10 Transitivity, Clustering coefficient and Redundancy

The clustering coefficient is defined as the proportion of paths of length two that are closed in a network and is expressed in Equation 1. It measures the degree of transitivity in a network.

$$C = \frac{(numberoftriangles) * 6}{(numberofpathsoflengthtwo)} \tag{1}$$

Using the transitivity() function we may calculate the probability that the adjacent vertices of a vertex are connected. For this graph, the average transitivity was 0.633.

We may also compute the clustering coefficient for a particular vertex i which equals to the number of linked pairs of neighbors of i divided by the total number of connected pairs of neighbors of i. Using the node with the greatest degree (20), node number 4, as an example, we can compute the cluster coefficient and obtain the result 190.

The redundancy is the mean number of connections from a neighbor of i to other neighbor of i.

3.11 Similarity, assortative mixing

There are many ways of defining similarity of nodes in a network, whether because they share the same neighbors (structural equivalence) or they have neighbors who are themselves similar [4]. When considering structural equivalence measures there are several methods that may be applied, including the the Jaccard Coefficient, Cosine similarity among others.

For example, we calculate different measures for the similarity between node 1 and 13, as is shown in the following table. We may use the similarity() function and define the method as "Jaccard" to obtain a matrix of the similarities between all nodes in the graph. There are no built-in functions for cosine similarity, Pearson Coefficient, and Hamming distance, but their formulae can be used to calculate these matrices.

Jaccard	Cosine	Pearson	Hamming
0.75	0.94	0.70	6

Assortative mixing corresponds to the tendency people have of associating with others they perceive as similar to themselves. In the case of this animal social network, it would be very interesting to obtain some characteristics of the tortoises, for example, their gender or age, to evaluate the assortative mixing.

3.12 Clustering Methods

Community detection is important to study networks, as well as separate them into smaller subsets. The goal is to detect natural divisions so that there are many nodes within the groups and few between groups. There are many ways to do this, one of which that corresponds to modularity maximization, meaning, when most connections between nodes of the same type (or in the same cluster).

In this project six different methods were chosen and implemented. The first corresponds to cluster_optimal, and is shown in Figure 10, a cluster algorithm that tries to find the maximum value of modularity. The value of modularity obtained was 0.473.

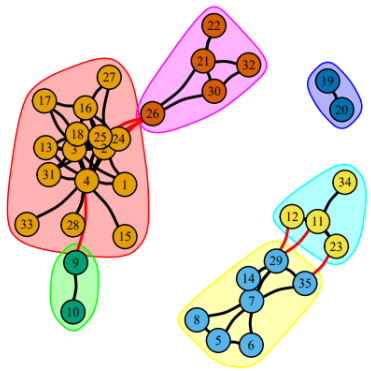

Figure 10: Optimal Clustering

Eigen clustering was implemented and the result is similar to the the optimal clustering, except that the second biggest component is all in the same cluster using eigen clustering (Figure 11). The modularity was equal to 0.471, slightly lower than the obtained with the previous method.

The function cluster_louvin() may be used to build the multi-level modularity optimization approach for discovering community structure. The result of applying this method is shown in Figure 12, showing a structure composed of 6 clusters, and the modularity obtained was 0.468.

Three clusters constitute the biggest component. One of these clusters overlaps with the region where cliques and cores were discovered, as well as the nodes with the highest centrality. In contrast, the smallest component consists only a single cluster.

Also, information theory was used to create clusters. As we may notice, a much larger number of clusters was created using this method (Figure 13), for a total of 10 clusters. This negatively affected

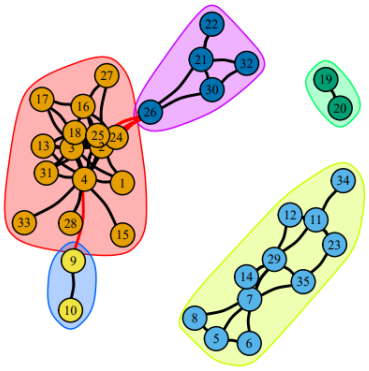

Figure 11: Eigen Clustering

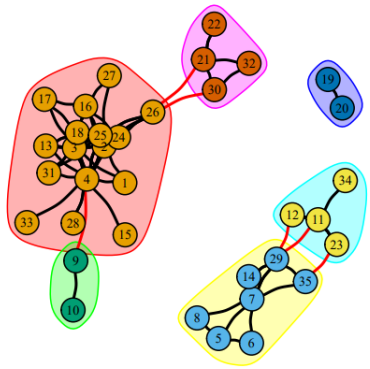

Figure 12: Clustering with the Louvin Method

the modularity value, which was the lowest for this clustering method, corresponding to 0.297.

Another potential approach to this problem is to identify the network edges that connect communities. If we can locate and eliminate these margins, only isolated communities will remain [4]. This is what betweness clustering is based on. This method was implemented and a modularity of 0.466 was achieved.

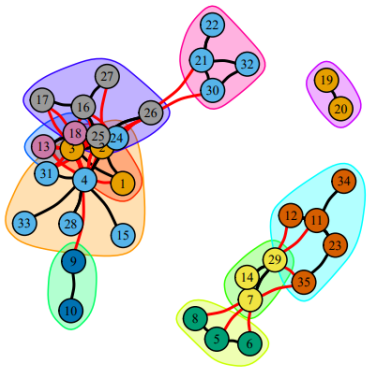

Figure 13: Information theory

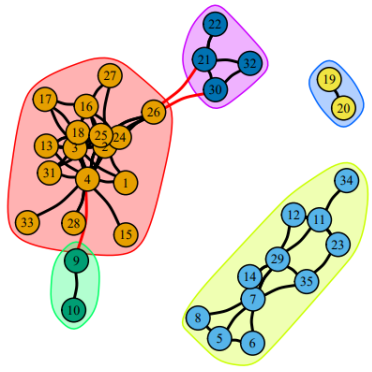

Figure 14: Betweness Clustering

For the hierarchical clustering, firstly we must find the best number of clusters, as the algorithm needs that the number of clusters are explicitly given. For this, we may see the distances (calculated by the complete method) between the clusters in the following dendogram (Figure 15), and choose a cut that will maximize the distances between clusters.

After observing the dendogram, we may realize that 3 seems to be a good number of clusters. For that reason, hierarchical cluster using 3 clusters and the Jaccard method was implemented, and is shown in Figure 16. The value of modularity obtained was 0.403.

As is expected, the optimal cluster method maximizes modularity. Since this graph has a small number of nodes and edges, this is the best clustering algorithm. Nevertheless, since modularity optimization is an NP-complete problem, and all known algorithms for it have exponential time complexity. This means that this function may not be useful on larger graphs, as it will take a long time to produce a result.

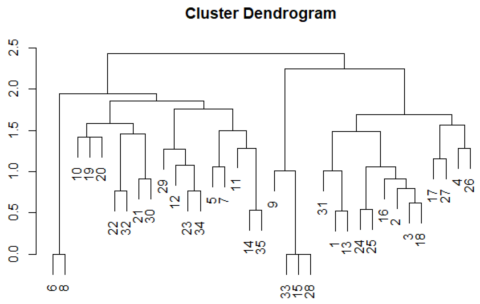

Figure 15: Dendogram

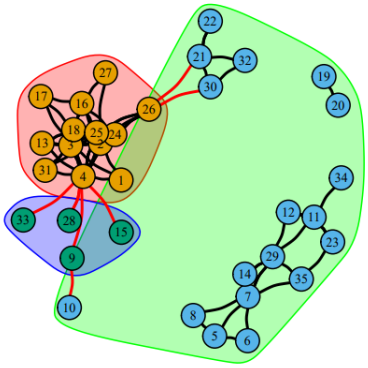

Figure 16: Hierarchical Cluster

3.13 Comparing to real world networks

In order to make a comparison with other graphs concerning biological networks, some examples from the recommended book [4] were considered. As we may notice, the biological networks concerning animal interactions are usually undirected and sparse, varying from tens to hundreds of individuals with hundreds of connections, as is the case of out network. The mean degree is similar to our data set, varying from 4.46 to 10.84. Nevertheless, this network shows a different component distribution, with a much lower fraction of nodes in the largest component. It is not possible to calculate the mean distance of all nodes in this network as it has many components.

4 Conclusion

The chosen graph for this research allows us to observe the interactions between tortoises through time. The investigation revealed an undirected, unweighted multi-graph with 35 nodes and 105 edges.

The research on degree distribution revealed that it is not possible to confirm that it follows the power law. In addition, the sparseness of the network and the distinct components that comprise it lead us to believe that this is not a unique tortoise community.

Different centrality measurements demonstrated that certain tortoises are more central to this network

(independently of the method). These zones of greater centralization are comprised of cliques and cores. Six clustering algorithms were implemented and the Optimal Cluster as shown to be the best in maximizing modularity while not being extremely time consuming.

References

[1] Wey, T., Blumstein, D., Shen, W., Jordán, F.. Social network analysis of animal behaviour: a promising tool for the study of sociality, Animal Behaviour (2008) https://doi.org/10.1016/j.anbehav.2007.06.020.

[2] Rossi, R. and Ahmed, N. The Network Data Repository with Interactive Graph Analytics and Visualization (2015)

[3] Run R file handed with the report

[4] Newman, M. Networks. Oxford University Press. Second Edition (2018)

GRAFICO DA COMULATIVE DISTRIBUTION DOS DEGRESS