Machine Learning in Data Science Project Week 11

Eva Aßmann, Paul Vogler, Katrin Böhler

Januar 2020

1 Network Statistics

In this task, fish trade data from 1998 which was published by the UN was analyzed using statistical network analysis. The data comprised two files, one containing the names and IDs of 151 countries that traded fish with each other, and the other file storing information on every export/import pair and the transaction value in million dollars. The files were read into a directed, weighted graph using the NetworkX network analysis framework, such that the nodes represented the country IDs with the countries names as attributes, the 2.744 directed edges (u,v) corresponded to country u exporting fish to v and the edge weight storing the transaction value. The analysis steps performed on the graph and the respective results are described below.

a)

The fish trade network's largest strongly connected component comprised 130 nodes, which is around 86,09% of all nodes.

b)

Yemen was observed to be have the longest distance from Germany regarding both import and export. The longest distance between Yemen and Germany was computed based on all weighted shortest path length with Germany as source and target. The resulting longest distance was 4,078.

c)

46,72% of the network's edges showed to be reciprocal. The reciprocity of a directed graph describes the ratio of the number of edges pointing in both directions to the total number of edges in the graph. In this context, 46,72% of all edges in the network connect countries that have conducted both import and export with each other.

d)

Looking at the in-degree of all nodes, i.e. the number of countries every country imported from, and sorting them in a descending order yielded the following top 10 fish importing countries in the network: Japan was observed to have imported fish from the highest number of countries in 1998.

Country X	Number of countries that X imported from
Japan	95
USA	92
France,Monac	91
Germany	85
Spain	83
Italy	81
UK	76
Netherlands	75
China HK SAR	70
Belgium-Lux	68

e)

The export revenues for each country were obtained by summing up the weights of each node's outbound edges. The five countries with the highest export revenues in the network are listed below: Thailand showed

Country	Export revenue in mio. dollars
Thailand	4046.18
Norway	3513.28
China	3059.8
USA	2553.94
Canada	2474.06

to have exported fish for the highest gains in 1998.

f)

For every node, the difference between unweighted in- and out-bound edges was calculated. The node with the highest difference of 53 was Morocco, having imported from 5 and exported into 58 countries in 1998.

2 Network Models

a)

An undirected graph with 20000 nodes was generated by starting with 3 nodes with a triangle of edges between them. After that each new node n was either connected to a uniformly at random chosen node o of the graph with probability p, or to to a neighbour of o with probability 1-p. The normalized degree density of generated graphs with p=0, 0.2, 0.4, 0.5, 0.8, 1 was plotted in a log-scaled graph seen in Fig. 1.

b)

To follow a power law distribution, the data points need to form a linear relation in the log-scaled plot. For the probabilities of 0 to 0.5 the points are partially linearly aligned, but all values with probability close or equal to 0 distort the linear relationship. The degree distribution for p=0.8 is the one closest to a linear relation and therefore the one that is probably the best for fitting a power law curve to. For p=1 the plot looks more like being exponentially distributed, where the curve curves downwards in a loglog plot.

c)

Now a power law line is fitted to the points from exercise a) with probability 0.2. For this the library powerlaw was used, that uses the probability values obtained from the normalized degree distribution to fit a

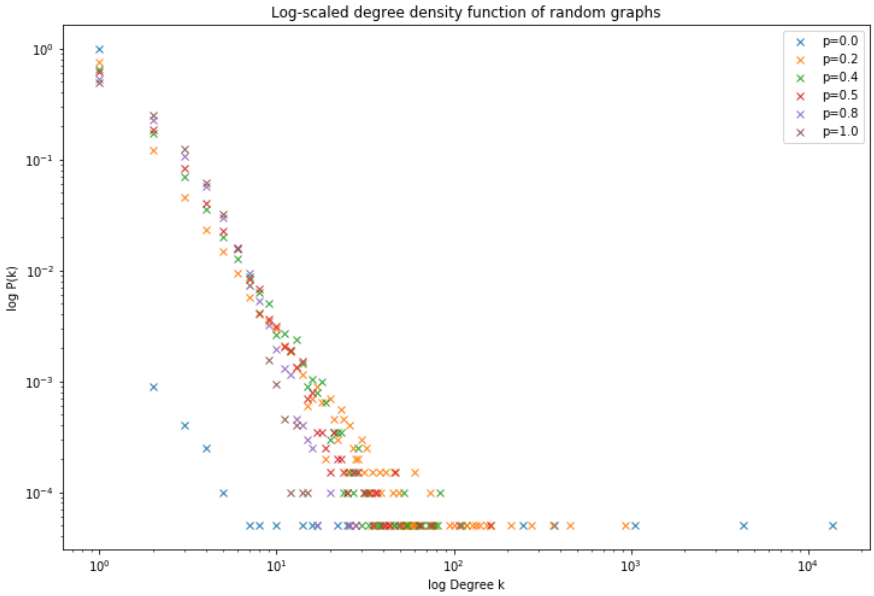

Figure 1: Normalized degree distributions of random graphs from 2a) in a loglog graph for different p.

linear function through the log-scaled input values (Fig. 2). The function returns the xmin and alpha values that define the power law function of $p(x) = (\alpha - 1)/x_{min} * (x/x_{min})^{(-\alpha)}$. The results were $\alpha = 1.529$ and $x_{min} = 0.00055$. As a measure of confidence of the fit the log likelihood between two distributions fitting to the data was calculated. The first distribution was the power law distribution and the second ones were the exponential, the lognormal and the stretched exponential distribution. The resulting likelihood value R is positive if the power law is the better fit for the data and negative otherwise. Additionally a confidence p is given for each of the likelihoods. The results are (R:3.6,p:0.0002), (R:0.088,p:0.93), (R:0.44,p:0.65) for the exponential, lognormal and stretched exponential distributions respectively. This shows that the power law distribution is the best to describe the data from the here tested heavy tailed distributions.

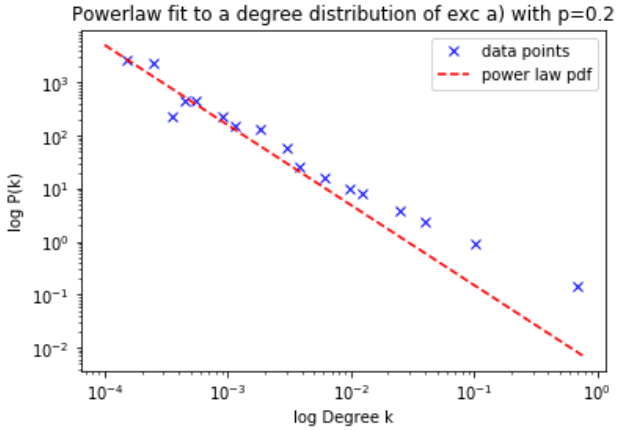

Figure 2: A power law function fitted to the normalized degree distribution for the graph produced in 2a) with p=0.2.

d)

A scale free graph was generated using the Barabási–Albert model with 20000 nodes and 400 connections for each node. The normalized degree distribution was plotted in a loglog graph and a power law function was fit to the points using the same function as above (Fig. 3). The results were $\alpha = 1.836$ and $x_{min} = 0.0001$. The fit was compared again to the same distributions as before, to get the resulting log likelihoods. The results are (R:13.87,p:9.54), (R:-1.867,p:0.06), (R:-1.47,p:0.14) for the exponential, lognormal and stretched exponential distributions respectively. So for these data points the lognormal and stretched exponential distribution seem to be better fitting to the data.
In comparison to the degree distribution from task c), the power law was a worse fit for this model than for the one in c).

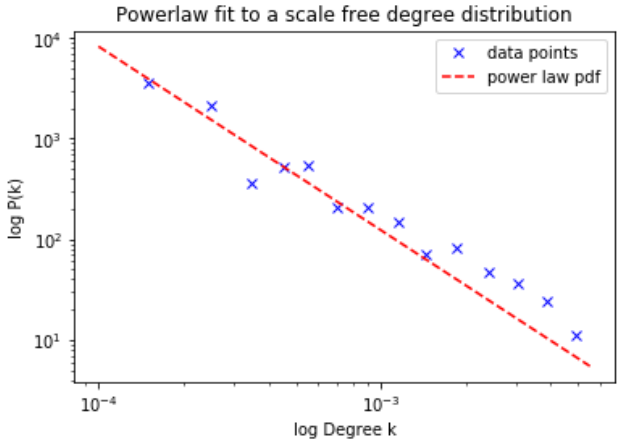

Figure 3: A power law function fitted to the normalized degree distribution for the graph produced in 2d).

e)

An Erdos-Renyi random graph with N=2000 nodes and probability p=0.1 was generated, without the help of an existing library function for it. The normalized degree distribution of this Erdos Renyl graph seems to be normally distributed around a mean of N*p (Fig. 4). The difference between this distribution and the one from task a) was, that in a) every new node could only be connected to a small subset of nodes that were already added to the graph beforehand. This resulted in a few of the initial nodes having a very high degree and the later ones having a low degree. In e) every node could be connected to any of the other nodes, regardless of order, which resulted in the normally distributed degrees.

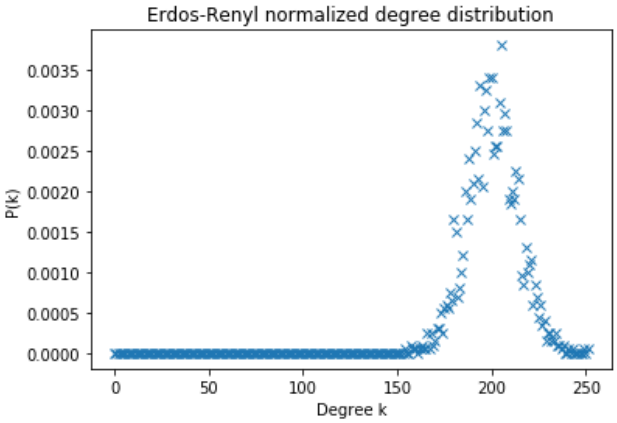

Figure 4: The normalized degree distribution for the graph produced in 2e).

3 Central Actors

For this task, a reduced version of the IMDB movies data set was used. This subset contained all actor-actor collaboration edges where the actors co-starred in at least 2 movies together between 1995 and 2004. In order to determine the most central actor, three centrality measures from statistical network analysis were employed: degree, betweenness and closeness. Two files were used to create a network graph: One file contained information on the actors node IDs in the network, their name, the number of movies they starred in, their main genre, all genres they once worked in as well as the respective number of movies within the respective genre. The other file contained the actor-actor collaboration pairs as well as the number of movies the respective two actors co-starred in. The data was read into a weighted, directed graph structure with the Networkx network analysis framework. The node IDs represented the 17.577 actors with name, number of movies, main genre and overall genre information as attributes. The 287.074 directed edges (u,v) represented a collaboration between actors u and v with the number of co-starred movies as weight. The following analysis steps were performed for the largest weakly connected component of the resulting network graph, which comprised 17.455 actors, i.e. 99.31% of all nodes.

a)

Degree centrality was calculated for all actors and the 20 actors with the highest degree were obtained. When looking at the number of movies made by these top 20 actors compared to the rest, it was observed that most of the top 20 actors made around 200 to 600 movies, while most of the remaining actors made between 10 and 100 movies (Fig.5). The high number of movies made by the top 20 stands in relation to the number of collaborations: While most the remaining actors worked in around 1 to 100 collaborations, most of the top 20 actors worked in 400 to 700 collaborations (Fig. 6). Since most movies show more than one single actor, the more movie an actor plays in, the more collaborations he has worked in. Thus, actors with a large portfolio of movies also have connected with a high number of other actors and have a high degree centrality within the IMDB network.

b and c)

Since the computation of betweenness and closeness centrality was very expensive and the first trials made Google Colab throw runtime errors, we started to do the computation locally with python. The centrality calculations took over 24 h and could not be saved because of a typo in the code. Due to having been occupied with the assignment for over 8 hours and the work schedule outside university, we did not manage to redo the calculations and perform the analyses in time. There are some analysis steps we planned on running on the centrality results that are written as commented code in the notebook.

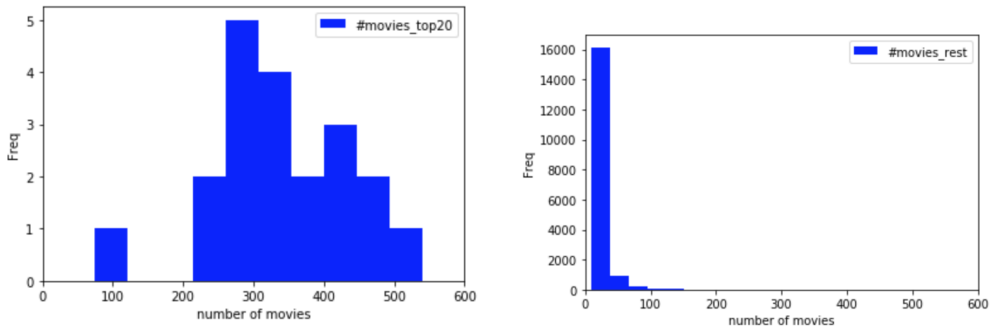

Figure 5: Comparison for number of movie from top 20 actors by degree and the rest

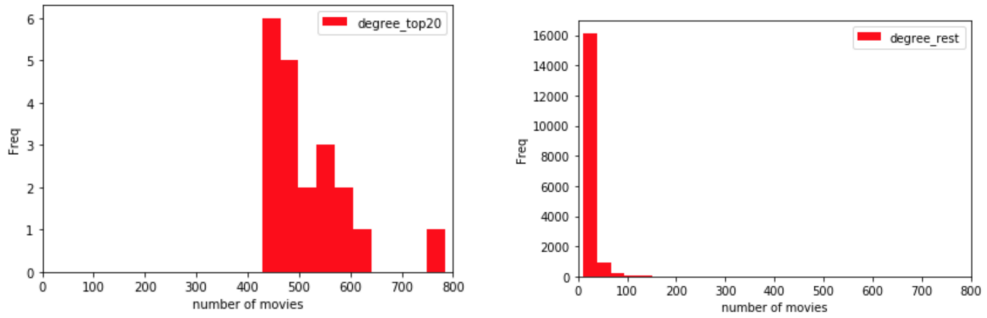

Figure 6: Comparison for degree of top 20 actors by degree and the rest

4 Machine Learning

a)

For predicting the links we chose a network that contains time-information. The network contains nodes, which represent tortoises and edges, which represent a social interaction between two tortoises. We read in two undirected graphs without weights, for the years 2008 and 2009. The first graph has 25 nodes, 11 edges and 27 connected components, while the second one contained the same nodes with 5 edges and 30 connected components. The 2008 graph was used to predict a link, while the 2009 graph was just used to compare the new link to. We created a training set, containing 80% of the edges of the 2008 graph.
For predicting future edges we need to measure similarities between pairs of nodes and then link the most similar ones. We did this with three different methods. The first one is the Jaccard Coefficient, this is a normalized common neighbors score. The ratio is calculated by dividing the number of common neighbors of two nodes by the number of the all the neighbors their union contains($S(i,j)=\frac{|N(i)\cap N(j)|}{|N(i)\cup N(j)|}$).
The second method we used to calculate similarities between two nodes is the Adamic-Adar Index. For each common neighbor of the two nodes we want to compare, we now sum up 1 divided by the total numbers of neighbors of that node. This way common nodes with lots of neighbors are lesser significant than nodes with elements shared between a small number of neighbors and these are weighted more heavily ($S(i,j)=\sum_{k\in N(i)\cap N(j)} \frac{1}{log|N(k)|}$).
The last score we used is called the preferential attachement. The probability that a new edge involves node i is proportional to $|N(i)|$, the current number of neighbors of node i. The probability of co-authorship of i and j is correlated with the product of the number of collaborators of x and y. Therefore we now calculate the score with the product of $|N(i,j)| * |N(j)|$.
For summarizing and comparing the results of the scores we calculated the AUC - ROC curves (Fig. 7). We

empirically chose the Jaccard score to be the best one, because all three methods showed the same AUC score. We then checked the scores made by the Jaccard scoring and the link with that score for predicting a new link. The maximal score achieved was 1, the link with that score was: (3,18). This connection seems very plausible because it was in the original 2008 graph and still persists in the 2009 graph, so this tortoise interaction seems to be an important one that consisted over more than one year.

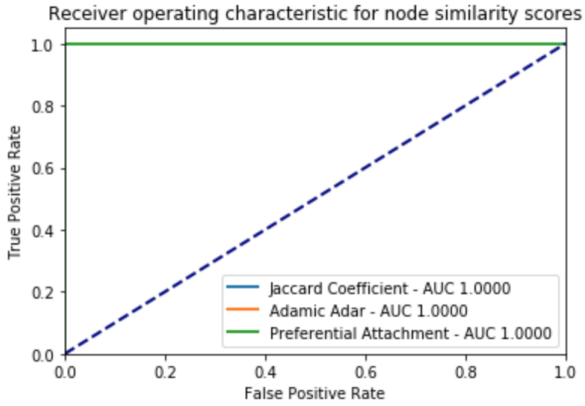

Figure 7: The AUC - ROC curves compared.

b)

In the second part we chose a network with known labels and removed some to predict those labels and check how good the prediction was. We have 123 nodes and 139 edges from enzyme data. First we attached the labels to the nodes. We have two different classes, and therefore colored the nodes separately in red and green. A numbered node belongs either to class 1 (red) or to class 2 (green) (Fig. 8). The two classes were not explained anywhere in the network repository. Afterwards we removed 70% of the labels for creating a test set. For label propagation we predict the label of a node by finding the label that is the most likely one. Nodes that are connected through high-density regions are likely to have similar labels. We use a prediction Matrix, that minimizes the smoothness and accuracy criteria. Therefore, there is always a tradeoff to make between the smoothness and the accuracy of our result. In the end we got a success rate of 88,62%. The results make sense respecting the method. Anyway, we do not think the score is good enough for using this prediction in real life on this network.

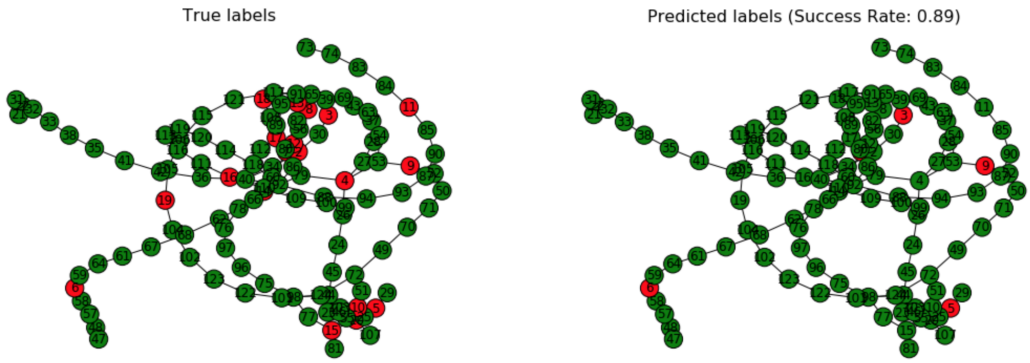

Figure 8: The network with the true and predicted labels.