Homework Network Science course - Part2

Cristina Gava

Matr 1153449

January 17, 2019

In this second part of the work I focused on the behavior of the network related to the community detection and prediction and ranking of nodes and links.

Link prediction

In the first part several link prediciton techniques has been applied, in order to see how the network responded: the methods used here are *Common neighbour, Adamic Adar, Resource allocation, Katz* and *Local Path*. First of all a certain amount of links has been removed from the network, in order to simulate a sort of temporal evolution of it and I then evaluated the performance of the algorithms. The removed edges have been:

- The links related to the highest degree nodes;
- The links related to the lowest degree nodes;
- Some random links;

In Table 1 the results are displayed: here the focus was on the value of *Precision factor* (out of all links, how many of them are predicted correctly), the *Recall factor* (out of all actual links, how many of them are predicted correctly), the *F-measure* (which helps to measure Precision and Recall at the same time using armonic mean and representing how much the two values are comparable) and finally the *AUC measure* (which indicates how much the model examined is able to distinguishe between the classes, in this case the true new links and the false ones).

Highest degree node removal				
Parameters / Methods	Precision	Specificity	F-measure	AUC value
Common neighbour	0.9303	0.9348	0.9639	0.7172
Adamic Adar	0.9035	0.9120	0.9493	0.4061
Resource Allocation	0.0590	0.5152	0.1114	0.1204
Katz	0.9705	0.9714	0.9850	0.8734
Local Path	0.9732	0.9739	0.9864	0.8003
Lowest degree node removal				
Parameters / Methods	Precision	Specificity	F-measure	AUC value
Common neighbour	0.9330	0.9372	0.9653	0.9477
Adamic Adar	0.9330	0.9372	0.9653	0.6708
Resource Allocation	0.9249	0.9302	0.9610	0.6700
Katz	0.9732	0.9739	0.9864	0.9673
Local Path	0.9705	0.9714	0.9850	0.9732
Random degree node removal				
Parameters / Methods	Precision	Specificity	F-measure	AUC value
Common neighbour	0.9410	0.9443	0.9696	0.8242
Adamic Adar	0.9276	0.9325	0.9624	0.8546
Resource Allocation	0.9223	0.9279	0.9596	0.8656
Katz	0.9732	0.9739	0.9864	0.9422
Local Path	0.9732	0.9739	0.9864	0.9323

Table 1: Comparison o the different link prediction algorithms wrt to three different types of removed links

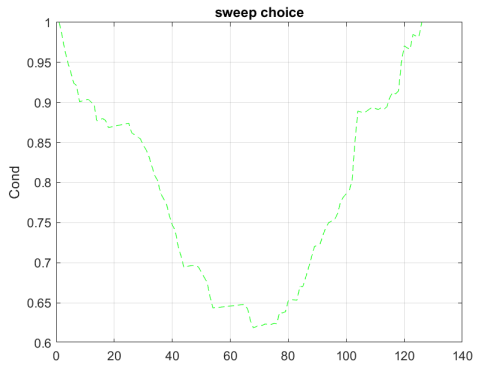

Figure 1: Conductance trend

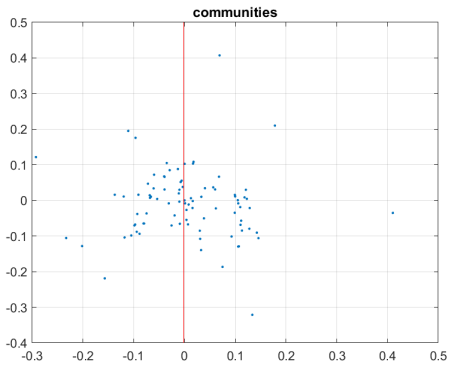

Figure 2: Communities separation

While the values of Precision, Specificity and F-measure are comparable in all the three cases and quiet good also, the more variability is present in the AUC value: it can be seen how Katz and Local Path methods always perform better than the other ones, while the Resource Allocation algorithm is always the one that performs poorly, compared to the others. If we then evaluate the AUC values with respect to the type of links removed, we can see an improvement in performance going from a lowest degree nodes link removal, to a highest one, passing through the random removal (in this case the function used was based on the generation of random values taken from a uniform distribution, so that the links were removed uniformly across the network). From these results we can conlude that in general the network behvaiour is predicted more precisely when it is composed by highly connected nodes. [1]

Community detection

For this part two approaches have been explored: in the first one a spectral clustering algorithm was employed, which took advantage of the Fiedler's eigenvetor (and so the minimum in the conductance values) to identify the two main communities of the network. The behaviour of the conductance is shown in Figure 1, where the minimum corresponds to the point over which is possible to perform the cut. The network so separated is shown in Figure 2 and consists of 2 communities of 68 and 59 elements respectively, with the minimum value for the conductance equal to 0.61863.

The second approach used the PageRank-Nibble procedure, where the conductance

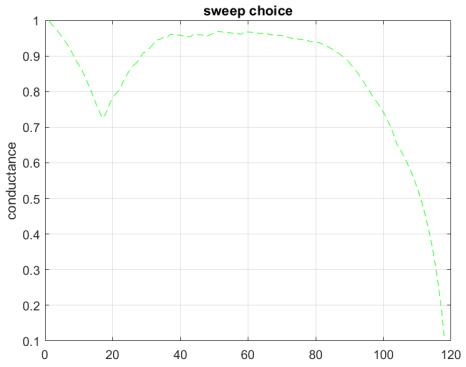

Figure 3: Conductance trend

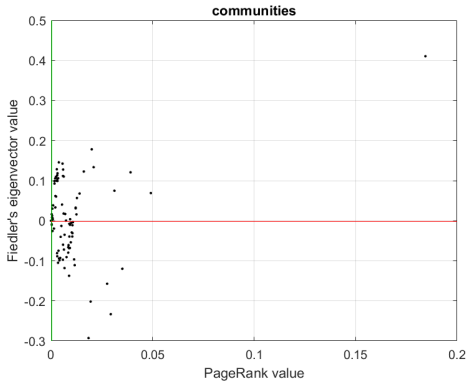

Figure 4: Communities separation

	Minimum conductance	Community1 size	Community2 size
Spectral clustering	0.61863	68	59
PageRank-Nibble	0.11111	118	9
Cheeger's bound		1.7493	

Table 2: title

value in this case was determined by the sweep over the nodes ordered according to the PageRank algorithm. In this case the communities found were definitely more unbalanced and the conductance value was quiet lower. The community separation is shwon in Figure 3 and Figure 4, while Table 2 summarizes conductance and community measures. Precisely, in the table also the Cheeger's bound is listed, and from tht it can be observed how both of the clustering options found are suitable in this case, since both of them have values for conductance $<$ than that bound. In particular, the second approach has the lowest value of it, and this could indicate that the corresponding community separation is the best one.

Ranking

Another part of the owrk was dedicated to ranking, with a major look to the PageRank and the HITS algorithm respectively.

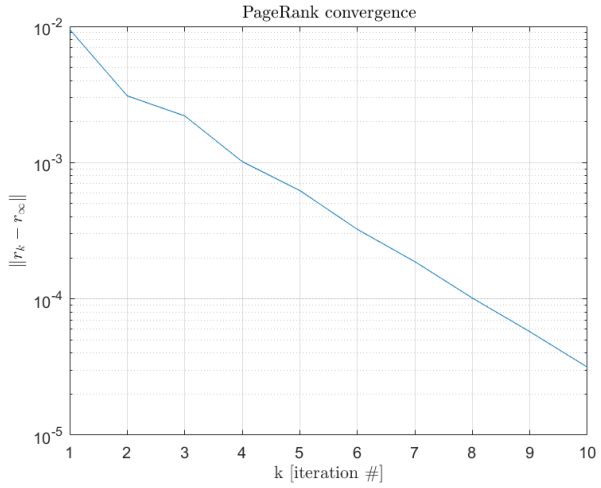

Figure 5: Convergence of the PageRank power iteration approach. From this it can be seen how the method converges quickly after a very small number of iterations

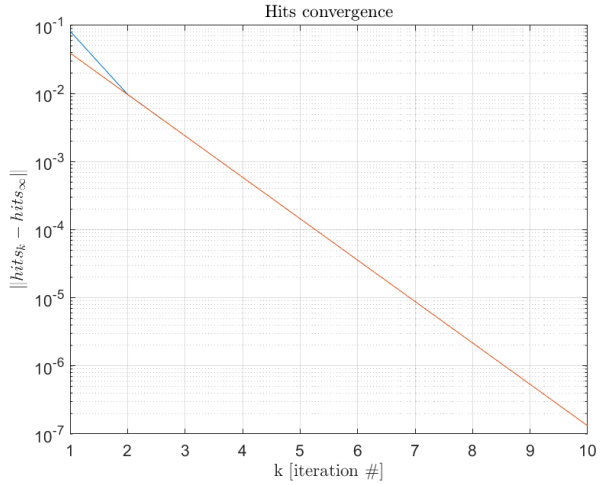

Figure 6: Convergence of the hits power iteration approach. This method is showed to converge approxhimately after 2 itertions

The PageRank algorithm

First the work the PageRank approach has been investigated in details, in particular the linear system approach has been compared to the power iteration one and an estimate on the number of iterations required for the latter to converge has been found. Figure 5

HITS algorithm

The other evaluation was made on the HITS approach, where the separation between hubs and authorities was not highlighted since the network we worked on is undirected. From Figure 6 it can be seen how also in this case the power iteration appraoch succesfully reaches

convergence after a very limited number of iterations, 2 in this precise case.

TrustRank algorithm

Finally, particular attention have been put also on the implementation on a TrustRank version, in order to evaluate the robustness of the network to spamming behaviours. The damping factor has been se to $c = 0.85$ and the number of power iterations to $M_b = 20$. Following the indications in [2], the algorithm has been developed: the first part consisted of the description of the $\sigma - function$ and the *Oracle function*, which were responsible, respectively, for the ordering of the nodes wrt their seed score and for the expression

$$\mathcal{O}(p) = \begin{cases} 0 & \text{if } p \text{ is trustworthy} \\ 1 & \text{if } p \text{ is untrustworthy} \end{cases} \tag{01}$$

that indicates whether a node can be considered good or not. The criteria selected to assign 0 or 1 to each seed was the belonging to one of the two communities found through spectral clustering. This supposition was based on the know idea that trustworthy nodes tend to connect/link to other trustworthy ones, while avoiding the spamming ones, in such way that the two categories can be seen as decently separated. The second part consisted in the actual implementation of the power iteration part, where we assumed a number L of seeds supposed to be scanned manually and whose value actually comes from the oracle function, while the rest of the scores are evaluated and saved in the vector $trust$. The final scores are then checked, to see if they are above or beneath a certain threshold, and so, if their reliability has been estimated accordingly to the belonging to one of the two communities.

Finally, from the trust percentages found, conclusions on the goodness of the approach can be drawn: indeed, there is a percentage of trust $\%_{trust} = 41.51\%$ (corresponding to the portion of nodes in the good community that are correctly estimated as good), and a percentage of distrust $\%_{distrust} = 75.68\%$. While the former value is not very good (more that 50%) of the good nodes are not seen as trustworthy, the latter is a good sign, since it shows that at least a great part of the bad nodes are recognized as such, concluding that the analyzed network is effectively scanned for "spamming" connections.

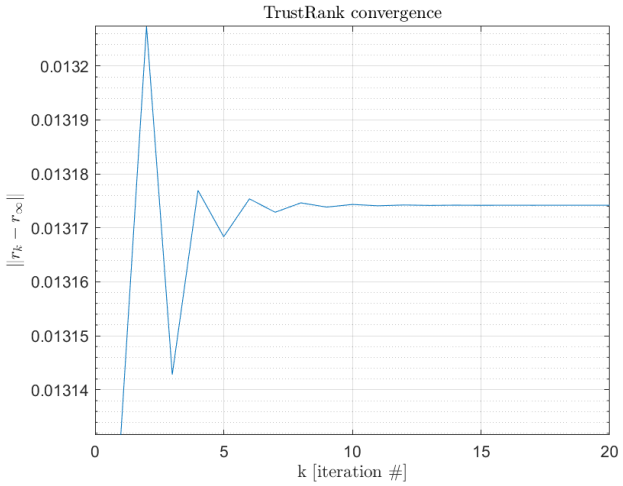

Figure 7: Convergence of the TrustRank power iteration approach. Also here, it can be seen how the method converges rapidly in < 10 iterations.

References

[1] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[2] H. Garcia-molina and J. Pedersen, "Combating Web Spam with TrustRank," 2004.