Lab Session 6: PageRank

Alex Carrillo Alza, Adrià Marcos Morales
Cerca i Anàlisi de la Informació, Ciència i Enginyeria de Dades (UPC)

October 23, 2019

1 PageRank

In order to compute the pageranks we are applying the recurrence studied in class $\vec{p}(t) = G^T * \vec{p}(t-1)$. This reccurence will not be applied in its matricial form since it can be optimized.

We know that $G = \lambda * M + \frac{1-\lambda}{n} * J$ where J is the matrix full of ones. So when we apply the recurrence $\vec{p}(t) = G^T * \vec{p}(t-1)$, we are computing $\vec{p}(t) = \lambda * M^T * \vec{p}(t-1) + \frac{1-\lambda}{n} * J^T * \vec{p}(t-1)$ which can be simplified since the sum of all pageranks is 1, so $J^T * \vec{p} = \vec{1}_n$ and $\vec{p}(t) = \lambda * M^T * \vec{p}(t-1) + \frac{1-\lambda}{n} * \vec{1}_n$.

With this simplified recurrence we avoid useless computations. Furthermore, since most of the values in M are 0, the $M^T * \vec{p}$ product can be optimized by storing the graph in its adjacency list form instead of the matrix form and computing the product only for those nodes in the list.

2 Experimenting

2.1 The appropriate damping factor

As a case of study, we try to check the assumptions made before regarding the behaviour of the convergence. It should be noted that the stopping condition parameter has been fixed ($\epsilon = 10^{-5}$) for all tests since it only influences the precision of the algorithm and therefore the number of iterations.

It is more interesting to see what happens if we change the value of the damping factor:

Damping factor	Top 3 ranked nodes	PageRank value	Iterations
0.2	DEN (Denver Intl, Denver, United States)	0.00215	5
	DME (Domododevo, Moscow, Russia)	0.00192	
	ORD (Chicago Ohare Intl, Chicago, United States)	0.00182	
0.4	DEN (Denver Intl, Denver, United States)	0.00365	7
	DME (Domododevo, Moscow, Russia)	0.00315	
	ORD (Chicago Ohare Intl, Chicago, United States)	0.00306	
0.6	DEN (Denver Intl, Denver, United States)	0.00489	10
	ORD (Chicago Ohare Intl, Chicago, United States)	0.00441	
	LAX (Los Angeles Intl, Los Angeles, United States)	0.00410	
0.8	DEN (Denver Intl, Denver, United States)	0.00593	20
	ORD (Chicago Ohare Intl, Chicago, United States)	0.00580	
	LAX (Los Angeles Intl, Los Angeles, United States)	0.00574	

Firstly, it is clear that as the damping factor approaches to 0, the Google matrix G loses the main pagerank values p_i given by M and leads to a solution closer to uniform. As a result, the algorithm converges faster and only takes few iterations for a guaranteed solution, but the outcome it is not interesting.

By contrast, as the damping factor approaches to 1 (changes are visible at 0.6) the solution becomes closer to the "true" pagerank, at the expense of a few more iterations. From the value 0.8, the accuracy is quite

good and the top ranked nodes make sense. In addition, the increase in iterations it is not a problem since there are no significant differences in execution time (0.238s vs 0.964s with damping factors 0.2 and 0.8, respectively).

With these results, we conclude that the appropriate damping factor should be between values 0.8 and 0.9, the same values we saw in Theory class as the common ones. However, in the next section, we are going to test some extreme cases with really high values.

2.2 The damping factor approaches to 1

Now we test which is the balance between speed and accuracy we can get. To be able to get a precise visualization of the behaviour, we run a round of 40 samples of damping factors between 0.8 and 0.995. As we know, the factor has to be $\lambda < 1$ to ensure uniqueness and (fast) convergence, although it is not in PageRank definition.

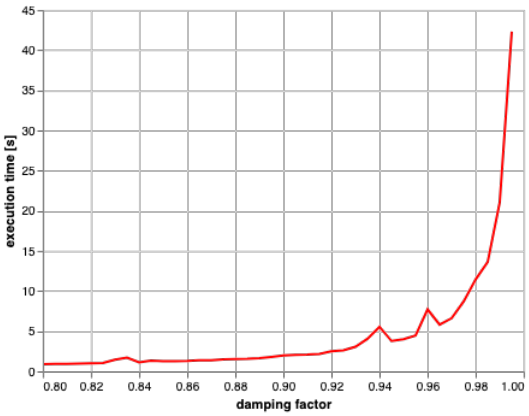

Figure 1: Behavior of the algorithm with high damping factors.

It can be observed that reasonable execution times are kept as far as a 0.98 damping factor, with some oscilations due to overhead on the way. From that point on, as we approach to 1 the algorithm takes an exponential time to converge, which is notable. We also notice that the top ranked nodes change at this point, getting the following result:

1. ORD (Chicago Ohare Intl, Chicago, United States)
2. LAX (Los Angeles Intl, Los Angeles, United States)
3. LHR (Heathrow, London, United Kingdom)

This experiment suggests that the damping factor should be tuned in every particular case, trying to be as close to 1 as possible but without compromising execution time.

3 Bonus Track

As the airport graph is symmetric we can apply a interesting property of symmetric graphs. A symmetric graph is a graph where $\forall (a,b) \in V,\ (b,a) \in V$. This means that a vector built as $v_i = |\{(i,j) \in V\}| = out(i)$ will be a eigenvector of eigenvalue 1 of the adjacency matrix, since we are sending $\frac{out(i)}{out(i)} = 1$ through each vertex, so each node will receive $in(i) * 1 = out(i)$ (as $out(i) = in(i)$) which is the same value than v_i.

So, considering that $\sum out(i) = 2|V|$, the Pagerank can be computed quickly as:

$$\vec{v} \mid v_i = \frac{out(i)}{2|V|}.$$