Khaoula Belahsen
khaoula.belahsen@yahoo.fr

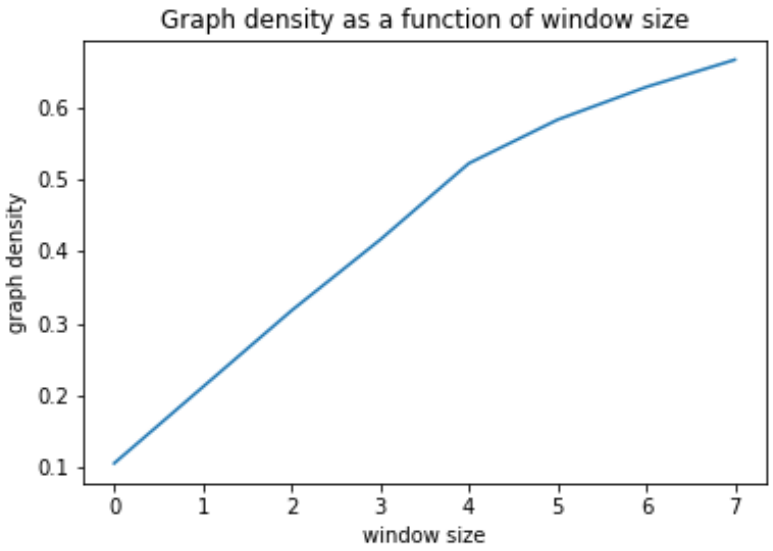
Lab session # 1
ALTEGRAD 2019

10/13/19

1 Question 1

Using the .density() function from igraph, I plotted the graph density as a function of window size, for values of window size ranging from 2 to 10.

The results are shown in the following plot :



We can see that the graph density increases accordingly to the increase of the window size. This observation is normal, including more words leads to more vectrices and edges.

We also observe that we have a piece-wise linear function with a change of slope at window size equal to 4.

- For smaller values of window size, we still have a general sentence construction with : subject, verb, complement for example. This results in a general information with more links between the words. Therefore, many edges between the nodes of our graph are created and the slope is very high.
- With a number of 5,6,7... of words in a window, we include more details of semantics. These additions are more specific to each sentence resulting in less edges created between the nodes. Therefore the increase of graph depth becomes slower.

2 Question 2

We define :

- $n = |V|$: cardinal of the set of vertices
- p average number of neighbors for each vertex

If we evaluate the algorithm's complexity line by line, we have :

Algorithm 1 k-core decomposition

Input: graph $G = (V, E)$
Output: dict of core numbers c

1: $p \leftarrow \{v : degree(v)\}\ \forall v \in V$
2: **while** $|V| > 0$ **do**
3: $v \leftarrow$ element of p with lowest value
4: $c[v] \leftarrow p[v]$
5: neighbors $\leftarrow \mathcal{N}(v, V)$
6: $V \leftarrow V \setminus \{v\}$
7: $E \leftarrow E \setminus \{(u, v) | u \in V\}$
8: **for** $u \in$ neighbors **do**
9: $p[u] \leftarrow \max(c[v], degree(u))$
10: **end for**
11: **end while**

- 1 : O(n) if immediate access to the degree of v, O(n*p) otherwise
- 2 : repeated n times
- 3 : O(n)
- 4 : O(1)
- 5 : O(p)
- 6 : O(1)
- 7 : O(p)
- 8 : repeated p times on average
- 9 : O(1)

So the overall time complexity for the algorithm is :
$O(n * p) + O(n * (O(n) + O(1) + O(p)) = O(n * p) + O(n^2 + n + n * p) = O(n^2)$

3 Question 3

I evaluated the performances of weighted and unweighted k-core decomposition against Pagerank and TF-IDF baselines. I obtained the following results where I tried 2 window sizes of 4 and 5 :

Metric	Unweighted k-core	Weighted k-core	Pagerank	TF-IDF
Precision	51.86	**63.86**	60.18	59.21
Recall	**62.56**	48.64	38.3	38.5
F1-Score	**51.55**	46.52	44.96	44.85

Table 1: Accuracy results with a window size of 4

Metric	Unweighted k-core	Weighted k-core	Pagerank	TF-IDF
Precision	52.16	**63.86**	59.19	59.21
Recall	**61.29**	48.08	37.7	38.5
F1-Score	**51.6**	48.08	44.26	44.85

Table 2: Accuracy results with a window size of 5

The precision from all 4 methods are in the range (50,64), with the highest precision being from the weighted k-core algorithm. Those of PageRank and TF-IDF were close to that maximum value, whereas the unweighted k-core was more far off from it.

However, the unweighted k-core performed best for values of Recall and F1-score metrics.

These k-core decomposition methods outperform the baselines PageRank and TF-IDF for the different values of window size.

4 Question 4

Advantages

- Because they are graph of words based methods, they take into account the dependency between different words.
- Fast and easy to compute
- Linear in the size of graph in best case and take only the main core as keywords instead of a percentage making them applicable to large scale data.

Disadvantages of unweighted k-core

- Doesn't take into account the co-occurrence between two words whereas the weighted version does.
- Provides less information than weighted k-core

5 Question 5

Possible improvements for the methods : [1]

- Compute an O(n) version of the algorithm : we used a very basic version
- Use heuristics such as dens or inf to select the number of core with respect to the desirable cohesiveness :
  - dens : go down the number of k-cores until density drops
  - inf : go down number of k-cores until shell size changes slope
- Use CoreRank to select the number of cores describing the cohesiveness (similar to dens and inf)

References

[1] Vazirgiannis Michalis. Graph-based text mining - 1. pages 6,24,58. DATA SCIENCE AND MINING GROUP, LIX, École Polytechnique, 2019.