Finding Key Articles in a Keyword Search

Hrishi Dharam, Eilam Levitov

March 16, 2017

Abstract

We began this project with a general goal to rank the 'fakeness' of news articles. During construction the goal has shifted slightly, but the concept is still similar. Given a search value, our program will return results corresponding to the 'center of mass' of each Voronoi region, or colloquially the most important result for a given group.

1 Introduction

In today's world there is a massive influx of 'hard to distinguish' data. The project's objective is to use Google's search engine to return the most prominent articles with respect to distinct 'groups' of articles. Here, instead of finding the most relevant and reputable articles, we construct a graph using the retrieved data and try to find communities within the graph that correspond to similar articles. By finding the centers of these communities, we can generate a list of articles that will summarize different topics related to the query.

2 Methods

2.1 Theory

To establish a logical relationship among the articles, we construct a fully connected markov chain where the transition probabilities correspond to the similarity between nodes (which correspond to articles). After quantizing the similarity and normalizing, we can use the PageRank algorithm in order to rank the similarity between all articles. Vanilla pagerank will allow us to rank articles by how similar they are to eachother but does not encode a notion of similar communities within the graph.

To achieve this we use an approach called personalized pagerank which allows us to bias pagerank results by a preference vector. At every timestep in the random walk interpretation, we will teleport to a state in the preference list according to some parameter α. Given that we are teleporting, the distribution of states we end up in is given by the preference vector.

A notion of distance between nodes is helpful to create a K-means style approach. The pagerank distance with teleportation constant is defined by Chung and Tsiatas on page 3. Pagerank-variance is given on page 4 is a measure of the variance between the personalized pagerank vector for a specific node (here the preference vector is just a one at the index of the node) and the nearby centers. A small pagerank-variance mean that the pagerank vectors are a good approximation for the centers of masses and indicates cluster formation. Cluster-variance is also defined on the same page and measures large discrepencies between personalized pagerank vectors for a specific node and the stationary distribution. A large cluster-variance indicates a big difference between the pagerank vectors and stationary distribution, indicating the formation of clusters. The main idea behind the algorithm to compute clusters, given on page 5, is that we want to minimize pagerank-variance while maximizing cluster-variance.

2.2 Pseudo code

```
1. Enter Search Value
2. Uses google to retrieve results
3. Parse text for each result
4. Generate a graph where edge weights correspond to the similarities of the documents
5. Adjust edge weights and apply threshold on similarities in order to transform to adjacency matrix
6. Transform the adjacency matrix into a transition matrix
7. Select parameters by simulations and apply the PageRank-ClusteringA [1]
8. Output the articles corresponding to the centers
```

3 Experiments

Please see Appendix: FakeRank.ipynb

4 Results

Our results are hard to interpret. There is a wide variety in the articles that are generated, demonstrating that the algorithm activly selects for dissimilar clusters. However, the results are difficult to evaluate without a prior expectation of the kinds of articles you expect to see. In the case where we queried "Trump, Russia," we had a selection of very different articles dealing with the a secret server scandal, the testimoney of FBI director James Comey etc. It is hard to be objective about which articles should be selected by the algorithm but we did see a reasonable number of expected topics

5 Limitations

Applying this algorithm to something subjective like news topics makes it difficult to evaluate as detailed in the previous section.

Our naive implementation of Personalized pagerank does not utilize more complicated linear algebra detailed in [Jeh 5] to improve runtime. We also didn't implement voronoi diagrams. A powerful visualization where articles are sorted by their pagerank distances (which should correspond to their similarities) should be able to give a much clearer picture of what the algorithm is actually doing. Furthermore, the way we created an adjacency matrix by setting a threshold similarity prevents us ffrom

6 Reference

1. Chung, F. & Tsiatas, A. Finding and Visualizing Graph Clusters Using PageRank Optimization.
   Internet Mathematics 8, 46–72 (2012)

2. Jeh, G. & Widom, J. Scaling personalized web search. Proceedings of the twelfth international conference on World Wide Web - WWW '03 (2003). doi:10.1145/775189.775191

3. PageRank Algorithm, 1998; Brin, Page. SpringerReference$doi : 10.1007/springerreference_57796$

4. Yadlowsky, S., Nakkarin, P., Wang, J., Sharma, R. & Ghaoui, L. E. Iterative Hard Thresholding for Keyword Extraction from Large Text Corpora. 2014 13th International Conference on Machine Learning and Applications (2014). doi:10.1109/icmla.2014.101