Weimar Bauhaus University

Faculty of Media

Big Data Architectures for Machine Learning and Data Mining

AuthorRank

Author:
Jiaqi Weng

Matriculation Number:
115131

September 7, 2017

1 Introduction

The challenging task of the seminar – "Big Data Architectures for Machine Learning and Data Mining" is to assess the importance of scholarly authors by using data from the Microsoft Academic Graph, which is a large graph containing scientific publication records, including citation relationships between publications, authors, institutions, journal and conference venues. An author is a person who publishes papers in some venues. A good author have two important characteristics. First, the author should have many publications, then he should have lot of citations for his papers.

In this paper, I will proposes an framework for ranking authors in the graph connecting the authors and the citation relationships of papers. In my approach, I will create a graph of citation relationship between authors based on the papers' citation network, and then, use a weighted-pageRank algorithm based on Google's Pagerank algorithm which ranking the webpage to calculate the rank of authors.

The remainder of the paper is organized as follows. I will introduce my proposed solution in Section 2. Then, the experiments and results will be discuss in Section 3. Finally, Section 4 concludes the paper and provides future work.

2 Methodology

In this section, I first introduce the structure of Microsoft Academic Graph followed by the process of framework and each steps.

2.1 The Graph Structure

In June 2015, Microsoft Research released a snapshot of its scholar data, namely the "Microsoft Academic Graph (MAG)"([1]). The large heterogeneous graph is comprised of papers, authors, venues (e.g., journals or conferences), affiliations, and the fields of study (Figure 1). The relationships and the inherent heterogene-

ity of objects in this graph provide new opportunities to evaluate and rank objects.

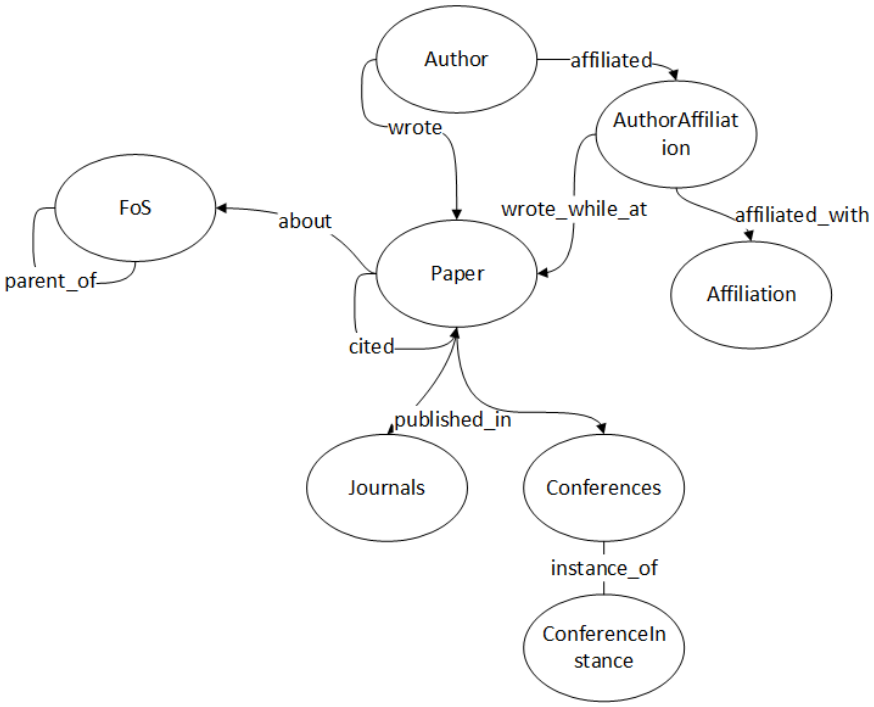

Figure 1: Academic Entity Relationship Graph

In addition, Table 1 shows the approximate counts of entities that we have in the resulting heterogeneous entity graph based on the snapshot taken in mid January, 2015([2]). In my project, I only focus on the subgraph which contains the citation relationship and paper-author relationship.

2.2 Author Ranking Framework

In this section, I will introduce the approach of the author ranking. Figure 2 represents the process of author ranking algorithm. It contains 4 steps: preprocessing, creating graph, computing rank and analysis of results.

Entity name	Entity Count
Papers	$>$ 83 million
Authors	$>$ 20 million
Institutions	$>$ 770,000
Journals	$>$ 22,000
Conference series	$>$ 900
Conference instances	$>$ 26,000
Fields of study	$>$ 50,000

Table 1: Counts of various entities in MAS corpus.

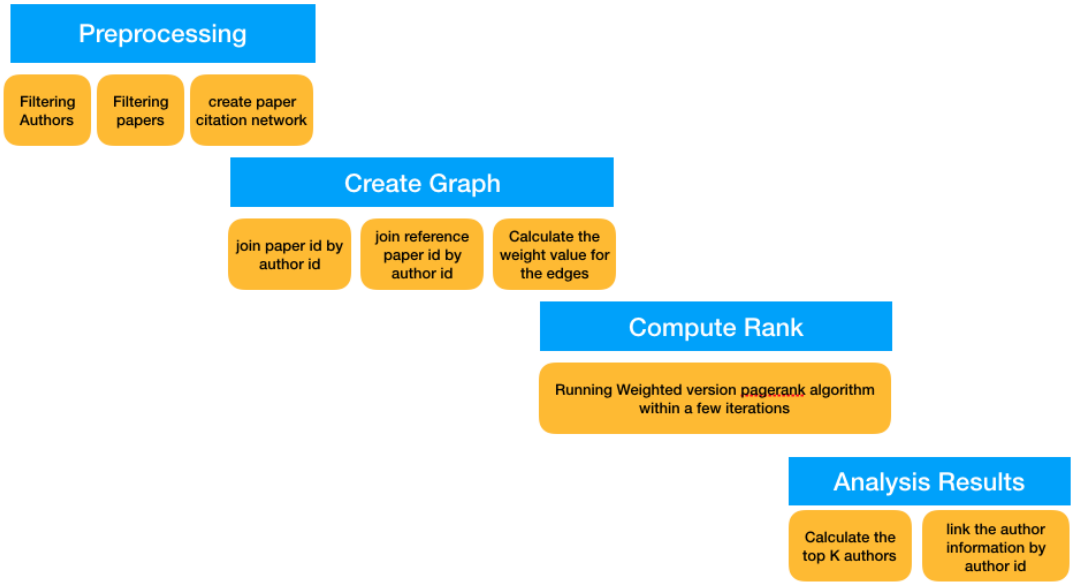

Figure 2: The workflow of Author Ranking

The development and runtime environment is using Hadoop MapReduce which is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner([3]). A MapReduce job refers to two tasks: map and reduce job. Map job takes a set of data and converts it into another set of data, and reduce job takes the output of map as input and

combines those data tuples into a smaller set of tuples. The yellow node of the workflow(figure 2) above shows the map reduce jobs.

2.2.1 Preprocessing

In the graph, many authors only wrote few papers. Thus, I will remove this part of authors. The first job need to count the number of paper that author wrote. Then, using a threshold to remove the author which wrote few papers. The second job is that remove the nodes of papers which the author of paper is not included in the author list filtered in the first job. Finally, use the rest nodes of papers to create the citation relationship network.

2.2.2 Create graph

The idea of creating the author citation graph is simple. From the previous step, I have the citation relationship graph. In this step, I only should join the paper with their author for the citation relation graph of paper. The join operation is a definition from relational database. It combines columns from one or more tables by using values common to each. Figure 3 represents an example of author citation relationship graph. It is weighted directional graph; the node is author; the direction of edge means reference relation or cited relation; the weighted value means counts of citation between authors.

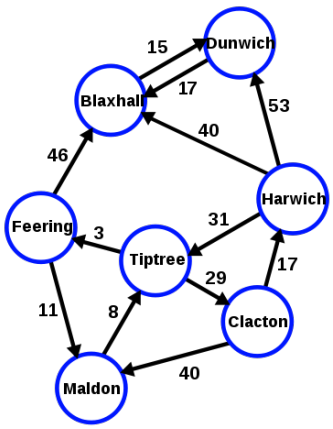

Figure 3: Example of author citation graph

2.2.3 Compute rank

From previous section, I get an author citation graph. And I will use PageRank algorithm to calculate the rank of the author from the graph. PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results ([4]). The normal form of the PR algorithm is given as formula 1. d is damping factor which means the probability that a user stop clicking the links of current page and request another random page when browsing a webpage and it usually set to 0.85. The p_j is the node from the set $M(p_i)$ of nodes linking to p_i. And $L(p_j)$ is number of links from p_j.

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \tag{1}$$

The weighted version of PageRank shows as equation 2. The weighted element in the formula is that weight of p_j to p_i is divided by the sum of weight of p_j to all nodes it links. The weight element replace $L(p_j)$ of the previous normal form.

$$WPR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M_{in}(p_i)} \frac{w_{ji}}{\sum_{p_k \in M_{out}(p_j)} w_{jk}} WPR(p_j) \tag{2}$$

As an example from the figure 3, the calculation of node Maldon is sum of value of two input nodes Clacton and Feering, and the weight element of these two nodes are 11/57(Feering) and 40/57(Clacton) respectively.

2.2.4 Analysis of result

The result of the ranking is also contains huge number of authors. Thus, I will use the Top K algorithm to filter the top k high ranking authors. Then I will preform jobs to get the information of the authors, e.g., author name, the number of papers, citation number. Finally, all the information will be linked to the author and create the author table as a final result.

3 Experiment result

Table 2 shows the final top 20 most important authors and their information.

Author name	Paper Number	Cited Number	Rank
walter c willett	2066	152804	0.0005357947
eric s lander	664	102834	0.0004255864
meir j stampfer	1317	114145	0.0004193795
mark j daly	719	121893	0.0003808914
bert vogelstein	712	77044	0.0003780531
graham a colditz	1268	97770	0.0003593641
shizuo akira	1305	71447	0.0003195555
vu	13658	91348	0.000296887
kenneth w kinzler	564	59975	0.0002954838
goncalo r abecasis	404	96204	0.000272488
charles h hennekens	744	65889	0.0002634507
david j hunter	1690	89573	0.0002616153
joann e manson	1391	78136	0.0002608663
david altshuler	346	83283	0.0002580041
david botstein	499	50945	0.000254834
eugene braunwald	1350	82927	0.0002544673
eric j topol	1527	84167	0.0002525223
karl j friston	990	58809	0.000251852
patrick w serruys	2433	101686	0.0002384653
francis s collins	782	68322	0.0002379901

Table 2: Final result of top 20 authors

In this section, I will introduce the configuration of the experiment. As mentioned in previous section, the dataset used is Microsoft Academic Graph. The tables of the dataset used in the experiment are PaperReferences.tsv, PaperAuthorAffiliations.tsv and Authors.tsv. The threshold of paper number setting for the experiment is 250. The author who wrote less than 250 papers would be removed

from graph. After filtering, there are 51226 authors left. The damping factor sets as 0.85. And the runtime of the process is not noted, it seems more than 3 hours.

4 Conclusion

In this work, I have proposed an weighted-PageRank algorithm and applied it to produce static rankings of authors in the Microsoft Academic Graph dataset based on Hadoop MapReduce framework. Furthermore, I have learned a lot from the seminar task, including what Hadoop framework and MapReduce is, how to deal with the big data under the framework, how to find our useful features for the ranking problem.

The current work only focus on the citation relationship between authors. And there are some future directions to work on.

1. Adding other elements to the the ranking of author, e.g., venues, field of study, co-citation relationship (two authors are related if they refer to the same paper)...
2. Extending the ranking algorithm to other entities of the dataset.
3. Improving the performance.

References

[1] Microsoft Research Blog(2015). Announcing the Microsoft Academic Graph: Let the research begin *https://www.microsoft.com/en-us/research/blog/announcing-the-microsoft-academic-graph-let-the-research-begin/*

[2] Arnab Sinha, Zhihong Shen,Yang Song, Hao Ma, Darrin Eide, Bo-june (Paul) Hsu, Kuansan Wang(2015). An Overview of Microsoft Academic Service (MAS) and Applications *http://dl.acm.org/citation.cfm?id=2742839*

[3] The Apache Software Foundation(2008). MapReduce Tutorial *https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html*

[4] Wikipedia PageRank *https://en.wikipedia.org/wiki/PageRank*