Progress Report: Ranking of Academic Papers

1 Terminology

We define two new metrics: the $j-index$ of a publication and the $s-index$ of a researcher. These metrics aim toward a more holistic measure of research output/impact, beyond mere citation counts and h-index.

2 Experimental Setup

We used a Google Cloud Platform instance with 4 CPUs and 52GB RAM from the us-east1-b zone with a hard disk size of 100GB to perform our experiments. As explained in following sections, even such scale did not eliminate the need for careful management of resources such as the adoption of chunking strategies for memory-intensive tasks.

3 Datasets

- We used the DBLP v10 dataset [1] which consists of over 3 million publications and 25 million cited-by relationships. This is our primary dataset and we have used it directly to construct the citation graph to calculate the reach function.

  From this dataset alone, we used for publications and researchers fields like: citation count, authors of a publication, the publication date etc.

- We used the Scimagojr journal and conference ranking dataset [2] in order to find the journal/conference rank (Scimago Journal Rank - SJR[3]) for a given publication. It contains data on almost 34k journals/conferences, almost 7k of them being in Computer Science. This dataset also contains the sub-fields within Computer Science that a particular journal can be ascribed to (such as AI, ML etc.). As explained in the following section, these factors are directly fed into the $s-index$ and the $j-index$ computation.

- Joining the aforementioned datasets was non-trivial as the join was to be performed on the venue column of the DBLP dataset and the Title column of the Journal Ranking dataset. Both of these columns are strings and after a vanilla equi-join, only 63k of out about 307k rows were matched. After employing a fuzzy-join using the fuzzymatcher[4], based upon the principle of Probabilistic Record Linkage [5], about 180k rows were reasonably matched (with a match score of 0.25 or above). An example of a match of score 0.25 is depicted in Figure. Fuzzy joining the two datasets was proved to be hard on memory, forcing us to join the two datasets in 11 chunks of size 300k rows each.

- The Open Academic Graph project, which is a project that links the Microsoft Academic Graph (MAG) and the AMiner datasets, proffers a much richer dataset, having over 166 million publications from MAG and 154 million from Aminer. These publications are not limited to Computer Science and are spread across a wide variety of disciplines. The dataset is feature-rich, having fields such as keywords, language, article length and the document type. Using this dataset instead of the DBLP v10 dataset proffers obvious advantages. However, the infrastructure required to carry out this task is considerable and well beyond the non-premium-plan limitations of the Google Cloud Platform. The dataset has an overall compressed size of 143.6GB. The Aminer data by itself takes up 39GB. We had considered using a subset of the Aminer data, to make the problem tractable in terms of memory and computational overhead. However, this approach has serious implications on the calculation of the reach function. This is attributable to the fact that for any given publication, the publication may be cited by another publication that does not reside in the subset that we have considered. This is why we decided to move ahead with the DBLP dataset, coupled with the Scimago Journal Rank dataset.

- We use the top-1000 ranked researchers in Computer Science by $h-index$ from the guide2research.com[6] for validation. We extracted the data from the webpage by a simple drag-and-select operation for each page of results (10 pages).

4 Ranking Method

First we calculate for every publication p, a citation score, which corrects for year bias. Year bias arises from some years being witness to more citations than other years. For example a relatively new paper could have a lesser number of citations because it is new. The citation score (C(p, y)) for a publication p published in year y is given by the formula:

$$C(p,y) = \frac{c(p)}{(C^m(y) \times C^t(y))}$$

where c(p) is the number of citations received since date of publication and $C^m(y)$ is the maximum number of citations any paper published in year y has achieved and $C^t(y)$ is the total number of citations for all papers published in year y.

For every publication, we obtain its SJR value SJR(p) as described in the Dataset section.

Finally, we also obtain the reach score(R) for every publication. This is explained in detail in the following section.

The j-index J(p) of the publication p is given by the formula:

$$J = (w_1 \times SJR^o(p)) + (w_2 \times C^o(p,y)) + (w_3 \times R^o(p))$$

where $SJR^o(p)$, $C^o(p,y))$, $R^o(p)$ are the scaled SJR score, citation score and reach score respectively. Scaling of all these fields follow the same strategy: divide each score by the maximum value of that score in the dataset. w_1, w_2 and w_3 represent weights for each factor. Currently, we have empirically determined the following weights:

w_1= 0.01, w_2 = 0.5 and w_3 = 1.

Then $s - index$ of a researcher r is computed as follows:

$$S(r) = \sum_{p \in p(r)} J^o(p)$$

where p(r) is the set of publications in which r is an author and $J^o(p)$ is the scaled j-index (scaled with the same strategy as above).

5 Reach

The "reach" of a work recognizes how fundamental it is. We consider a graph of citations, with the nodes being publications and directed edges representing the "cited-by" relationship. Figure 1 shows two examples of such a graph.

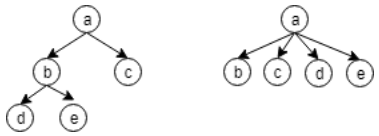


Figure 1: Two possible citation networks for publications a,b,c,d,e

We define the reach of a node p as follows:

$$reach(p, par, h) = k(par \rightarrow p) \times h \times |adj(p)| + \sum_{q \in adj(p)} reach(q, p, h+1)$$

The reach algorithm resembles the modified version of pagerank[10] algorithm, however, it only accounts for outgoing edges rather than a combination of ingoing and outgoing edges for each node as seen in traditional pagerank. This reach score is in turn fed back into the j-index ranking metric calculation as discussed in previous subsection.

The above function calculates the reach of node p with parent par (i.e there exists an edge $par \rightarrow p$) and adjacency list adj(p) on the h^{th} hop of a traversal of the citation graph starting from some starting node. This structure awards more weight/importance to a publication which is cited indirectly at a larger distance in the network graph. k is a multiplicative factor that influences this. k is defined as follows:

$$k(par \rightarrow p, h) = \begin{cases} |p.subf \setminus par.subf| * h, & \text{if } |p.subf| \neq 0 \\ h, & \text{otherwise} \end{cases}$$

where subf is the set of sub-fields, $|subf|$ is the cardinality of the set.

We collect the values of k in a dictionary by traversing each edge in the citation graph up front. Then we memoize the values of reach in a dictionary as well, as there are overlapping subproblems of the form: reach(p, h).

For memory tractability, we maintain a recursion depth cutoff d, till which we explore citation relationships in the citation graph. We executed the reach computation for d = 5, 6 and 10. We observed that a cutoff greater than 10 resulted in convergence of the reach score and impacted the usefulness of the reach feature.

6 Results and Validation

1) **j-index**: Figure 2 depicts the top 10 publications by j-index. We can see that our ranking has captured some prominent publications in the field. Since we assign a larger weight towards reach score (w3 = 1.0) as compared to citation score (w2 = 0.5) and SJR score (w1 = 0.01), our results favor publications with a high reach score. This is especially evident in the difference in j-indices of the 0th and 1st ranked papers "The Design and Analysis of Computer Algorithms" and the "Introduction to Decision Trees". The 0th paper has a j-index of 1.000412 and the 1st paper has 0.6666723. This is directly attributable to the huge difference in their reach scores. Also, the 2nd ranked paper has a much lower citation count but still has a significant reach - This seminal paper, Visual Learning and recognition of 3-D objects has "reached" across multiple sub-fields. These aspects show that the reach score is a clear differentiator.

We also see that the 8th ranked publication: "On Quine's Axioms of Quantification" by George Berry has a significantly lower citation count (50) and reach score (1.68e-13). However our model rewards the paper as in 1941, when the work was published, the total number of citations racked up by other works in Computer Science in 1941 amounted to 64. This work contributes to 50 of the 64. Clearly this is the highest cited paper from that year. This is why this paper has the highest citation score of 1.0. The bias against papers recently published also seems to be somewhat accounted for due to the presence of papers from the 90s in the top 10. This depicts that the citation score is also a differentiator.

Finally, the SJR score perhaps has a very small weight (w1 = 0.01), rendering it relatively insignificant. This is done on purpose. We wanted to see if one of the factors would be dominated as a result of our weighting scheme. Notice how the 1st and 2nd ranked publications have a very small difference between their reach scores and citation scores but a significant difference between their SJR scores (0.05 and 0.16 respectively). Had there been no dominance, these two publications would have had their ranks flipped.

	title	authors	venue	year	n_citation	sjr_score	reach_score	citation_score	j-index
0	The Design and Analysis of Computer Algorithms	[Alfred V. Aho, John E. Hopcroft]		1974	13227	0.018838	1.000000e+00	0.000447	1.000412
1	Induction of Decision Trees	[John Ross Quinlan]	Machine Learning	1986	19320	0.050355	6.661679e-01	0.000103	0.666723
2	Visual learning and recognition of 3-D objects...	[Hiroshi Murase, Shree K. Nayar]	International Journal of Computer Vision	1995	2736	0.166715	6.568129e-01	0.000003	0.658482
3	Example-based learning for view-based human fa...	[Kah Kay Sung, Tomaso Poggio]	IEEE Transactions on Pattern Analysis and Mach...	1998	2234	0.171497	5.570867e-01	0.000003	0.558803
4	Notes on Data Base Operating Systems	[Jim Gray]	Advances in Computers	1978	2727	0.014853	5.345290e-01	0.000048	0.534702
5	Support-Vector Networks	[Corinna Cortes, Vladimir Vapnik]	Machine Learning	1995	26114	0.050355	5.326541e-01	0.000032	0.533174
6	Implementing remote procedure calls	[Andrew Birrell, Bruce Jay Nelson]	ACM Transactions on Computer Systems	1984	2838	0.104188	5.258613e-01	0.000033	0.526920
7	Probabilistic Reasoning in Intelligent Systems...	[Judea Pearl]		1988	6589	0.018838	5.204852e-01	0.000024	0.520686
8	On Quine's Axioms of Quantification	[George D. W. Berry]	Journal of Symbolic Logic	1941	50	0.027605	1.686521e-13	1.000000	0.500276
9	The design and implementation of INGRES	[Michael Stonebraker, Gerald Held, Eugene Wong...	ACM Transactions on Database Systems	1976	539	0.053760	4.983320e-01	0.000011	0.498875

Figure 2: Top 10 papers by $j - index$

2) **Validation of the s-index:** Figure 3 depicts the top-10 researchers by $s - index$ and a comparison of the h-index and the $s - index$.

We see that Takeo Kanade and Andrew Zisserman feature in both top-10 lists. A few others have s-index rank close to the top 10.

Herbert Simon and Terrence Sejnowski, who have very high h-indices have low s-indices. While Herbert Simon[7] is a very well known economist and political scientist, Terrence Sejnowski[8] is a leader in the field of Computational Neurobiology. Their ample citations and reach outside Computer Science is not captured by our dataset.

Jiawei Han[9] is a stalwart in the field of Data Mining, with many books and stellar publications. However, his research is narrower and more focused, as depicted in his reach score of 0.176.

Azriel Rosenfeld [10] was a leading researcher can be described as the father Computer Image Analysis, with pioneering contributions across the field over 40 years. He wrote the first textbook in the field, was founding editor of its first journal and was co-chairman of its first international conference. He published over 30 books and over 600 book chapters and journal articles, and directed nearly 60 Ph.D. dissertations. This suggests a high reach score and citation score for a large number of his publications, which explains his rank.

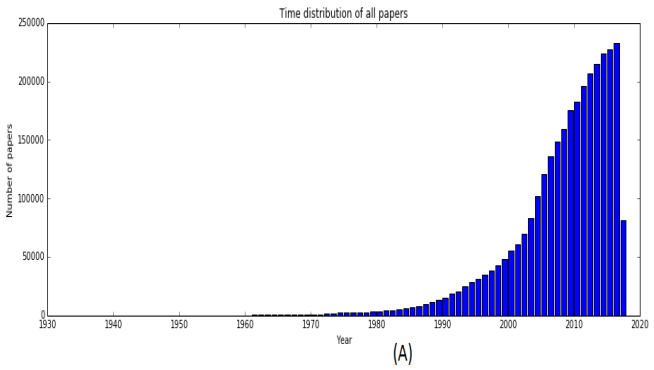
	authors	academic_age	s-index
0	Azriel Rosenfeld	42	4.014671
1	Michael Stonebraker	44	3.354656
2	Takeo Kanade	45	3.073051
3	Jitendra Malik	35	3.003419
4	Alex Pentland	36	2.953394
5	Andrew Zisserman	32	2.947706
6	Cordelia Schmid	25	2.597863
7	Pietro Perona	29	2.563672
8	Tomaso Poggio	32	2.539557
9	Robert E. Schapire	30	2.505502

(A)

	Name of Researcher	h-index	h-index rank	s-index rank	s-index
0	Anil K. Jain	179	0	27	1.712872
1	Herbert Simon	175	1	1230	0.227589
2	Jiawei Han	162	2	328	0.516504
3	Terrence Sejnowski	151	3	256	0.595427
4	David Haussler	151	4	13	2.265852
5	Takeo Kanade	151	5	2	3.073051
6	Philip S. Yu	148	6	198	0.680757
7	Michael I. Jordan	148	7	56	1.284312
8	Scott Shenker	146	8	102	0.995337
9	Andrew Zisserman	144	9	5	2.947706

(B)

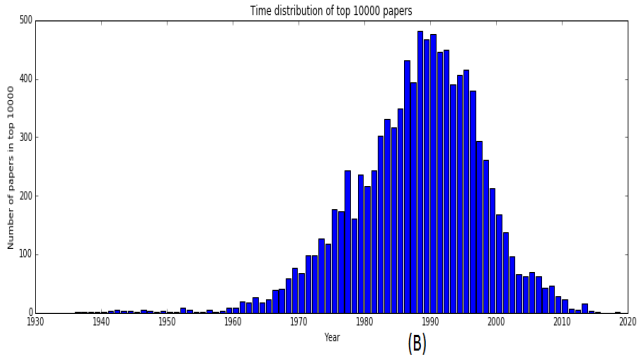
Figure 3: (A) Ranking of top-10 researchers by $s-index$. (B) Comparison of h-index and $s-index$ for the top 10 authors by h-index.



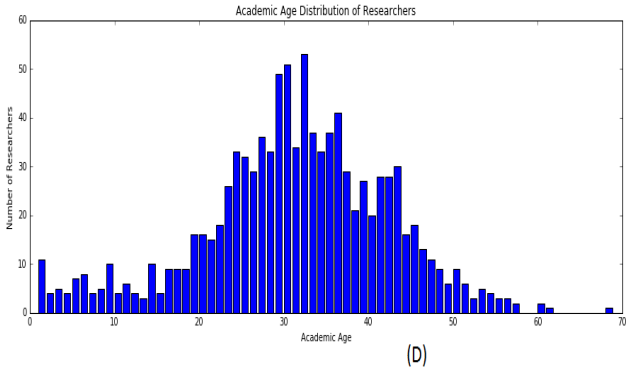
(A)

	sjr_score	reach_score	citation_score	j-index
sjr_score	1.000000	0.022276	0.000571	0.166167
reach_score	0.022276	1.000000	0.001642	0.975747
citation_score	0.000571	0.001642	1.000000	0.166137
j-index	0.166167	0.975747	0.166137	1.000000

(C)



(B)



(D)

Figure 4: (A) Distribution of papers by publication year. (B) Distribution of top 10,000 papers by publication year. (C) Correlation matrix of j-index factors. (D) Distribution of academic age of **top 1,000** researchers.

3) Our rankings and these time distribution plots (Figure 4A, 4B) show an important characteristic. Older papers usually have a lower citation count given their age and changes in their respective fields. To rise in rank, they have to be extremely fundamental, i.e. they have to have a very high reach. New papers have to be have a very high citation score on the other hand to deal with their inherently low reach (as there has not been enough passage of time for reach to develop). As the figure 4(B) depicts, research published in the 90s have the best ranks. They are at the sweet spot between reach and citation score.

4) From the correlation analysis (Fig 4(C)), we can see that the sjr_score, $reach_score$ and the $citation_score$ are not correlated

at all, showing that they are not derivative - i.e. the j-index is approached from completely different angles.

5) We can see that the academic age distribution of the researcher is normal-like (Fig 4(D)), emphasizing that an academic age of about 30 is the most common among top researchers.

7 Limitations

- Some publications don't have the actual complete list of references. Also, some publications don't have the actual complete list of authors. The classic work "The Design and Analysis of Computer Algorithms" by Alfred Aho, John Hopcroft and Jeffrey Ullman does not have Jeffrey Ullman listed as an author. This negatively affects the ranking of Jeffery Ullman.
- As described above, interdisciplinary researchers, who have publications across Computer Science and some other field take a hit in their rankings, simply because we don't have data on their publications in other fields.

8 Next steps

- We will scan the distribution of citations for a certain field by year for spikes and check in case of a spike, we see if our indices were inflated by the spike. If the inflation of our metrics exceed a threshold τ (empirically determined), we will attempt to reduce this inflation.
- To predict popularity of papers on arXiv, we will use a manually curated arXiv dataset as our test dataset and we will train a machine learning model on our existing data with the j-index as the y-label. If the index of an arXiv paper exceeds a certain threshold θ (empirically determined), we will consider it to be popular. We can readily extend this approach to identify researchers on arXiv who are to become popular. We can either construct this as a regression problem or a classification problem depending upon our target variable. The classification problem could be binary or multi-class. Possible machine learning models applicable here are Linear Model, Gradient Booster, and Random Forest.
- Improve our baseline model by correcting for field bias (certain fields have more publications than others). This can be easily performed by parameterizing $C(p, y)$ with f - the sub-field (AI, ML, Systems etc.) of the publication p. This means that we would have to constrain the calculation of denominators $C^m(p, y)$ and $C^t(t, y)$ by f.
- Improve our baseline model by tuning the weights for each factor in the $j - index$ calculation. As mentioned previously, our current weight assignment tests our hypothesis of dominance.
- Correct for the bias borne from the number of co-authors in a publication. Researchers often do not contribute equally to a publication. Also, authors of stellar publications should be awarded more if the number of their co-authors is relatively low. This data can be extracted with minimal effort from the dataset.
- Automate the validation of our ranking by joining the top-1000 researchers by h-index and our dataset of 1.77 million researchers. We have to employ a fuzzy join as the names of the authors across the two datasets are inconsistent.

9 References

1. DBLP v10 dataset - https://static.aminer.cn/lab-datasets/citation/dblp.v10.zip
2. Scimago Journal ranking dataset - https://www.scimagojr.com/journalrank.php
3. Scimago Journal ranking calculation - https://www.scimagojr.com/SCImagoJournalRank.pdf
4. fuzzymatcher library - https://github.com/RobinL/fuzzymatcher
5. Probabilistic Record Linkage - https://www.scimagojr.com/SCImagoJournalRank.pdf
6. Guide2Research h-index ranking - http://www.guide2research.com/scientists/ranking
7. Herbert A. Simon https://en.wikipedia.org/wiki/Herbert_A._Simon
8. Terrence Sejnowski-https://en.wikipedia.org/wiki/Terry_Sejnowski
9. Jiawei Han - https://en.wikipedia.org/wiki/Jiawei_Han
10. Pagerank - http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf